MoReTax

Handling factual information linked to taxonomic concepts in biology

MoReTax

Handling factual information linked to taxonomic concepts in biology

Contributors:
Walter G. Berendsohn, Markus Döring, Marc Geoffroy,
Karl Glück, Anton Güntsch, Andrea Hahn,
Regine Jahn, Wolf-Henning Kusber, Jinling Li,
Dominik Röpert, and Frank Specht

Department of Biodiversity Informatics and Laboratories, Botanic Garden and Botanical Museum Berlin-Dahlem, Freie Universität Berlin

Edited by Walter G. Berendsohn

Contents

The concept problem in taxonomy: importance, components, approaches

MARC GEOFFROY & WALTER G. BERENDSOHN [a]

Are names reliable keys for biological information?

Scientific names of organisms are commonly used as a reference system to organise biological knowledge. In order to ensure information quality when organism-referred data is represented and searched for in databases, the data must be assigned to a unique class of reference objects (taxa; e.g. botanical or zoological species) and this assignment must be stored permanently. Today, most databases try to achieve this by using the scientific name (e.g. species name). Names are therefore critical since they are the key for accessing information. Unfortunately, scientific names in biology are not well suited for this task, although there are strict rules concerning their application to a taxon ("Codes" of nomenclature, see SNEATH, 1980; FRANCKI & al., 1990; TREHANE & al., 1995; ICZN, 1999; GREUTER & al., 2000). In full accordance with the rules, different names may be applied to one and the same taxon. Naming at least of species and infraspecific taxa depends on their classification, for example, placement of a species in a different genus changes the species' name. Classification, in turn, is a matter of scientific opinion, which may be argued among peers and which is affected by the accumulation of new knowledge over time. The application of the same name to several differing taxa may be equally rule-abiding. Most Codes link the name to a type specimen, so that splitting or unifying taxa (which may be a matter of scientific opinion) results in different groups of organisms having the same name (because all of them contain the type specimen). As a result, different biologists can use the same taxon name without having the same opinion about which organisms belong to the taxon. Still worse, valuable information may be referenced by wrong names (not formed according to the rules of the Codes), or by misapplied names (good names used to denote the wrong taxon).

Different users of a name may have different concepts of what the name denotes. Different concepts can lead to different circumscriptions and therefore to different demarcations between taxa. Names alone therefore do not form a dependable index system, their use in a particular context must be considered as well. It is for this purpose that BERENDSOHN (1995) introduced the notion of "potential taxon", which identifies a taxonomic concept by referencing the context in which the name is used; e. g. *Hypnum flagellare* Dicks. sec. MÖNKEMEYER 1927. Similar notions were introduced as "taxon view" by ZHONG & al. (1996), as "circumscribed taxon" by PULLAN & al. (2000), as "taxonym" by KOPERSKI & al. (2000), as "taxonomic reference" by LE RENARD (2000), and as "assertion" by R. PYLE (after ANON. 2002). In the context of this publication, we will use the following terms:

- Taxonomic concept: set of explicit or implicit criteria, which allow to decide whether a particular element (specimen, observation or lower taxon) belongs to a taxon or not.
- Potential taxon: set of elements (disappeared, existing, not yet existing, or postulated) which fulfil the criteria of a taxonomic concept. The definition of an element here includes but goes beyond the one given for "instances (specimens or lower taxa)" in YTOW & al. 2001.
- Potential taxon name or taxonym: string concatenation of a scientific name, followed by "sec." [= secus, secundum, following, according to (STEARN, 1992)] and the

bibliographic citation of the source in which it was used. Used in order to designate a potential taxon and therefore also its corresponding taxonomic concept.

Taxonomic treatments have traditionally treated part of this problem by giving lists of synonyms (taxa whose type is placed within the taxon accepted in the treatment) and – sometimes – by reference to wrong or misapplied names. Within a single source, a certain overview of the historical and contemporary treatment of the taxon can be achieved. However, modern information systems increasingly attempt to unite data from different sources. It is obvious that the risk of concept instability grows with the quantity of different aggregate sources of information, but until recently no firm data were available to assess the extent of the problem.

This situation changed with the publication of the "Reference List of German mosses" (KOPERSKI & al., 2000). This is a pioneering work resulting from a research and development project funded by the German Agency of Nature Conservation, where for the first time concept orientation as laid out in the IOPI model (BERENDSOHN, 1997) was actually translated into a computer software used by taxonomists (GRADSTEIN & al., 2001). The data produced allow for an in depth analysis and statistical evaluation of the stability and instability of names and concepts, work in progress from which we here present some preliminary results.

Table 1: Basic data found in KOPERSKI & al. (2000)

Item	Quantity
Plant names treated (accepted and not accepted)	8.544
References (incl. the reference list itself)	12
Potential taxa (names in the context of their reference)	24.390
Names / concepts accepted according to KOPERSKI & al. (2000)	1.548
Explicit concept relationships (between accepted concepts and concepts from other references)	7.891

The Reference List consists of a conventional checklist of taxa accepted by the authors, representing their own taxonomic point of view, which includes a list of synonyms. The authors take a novel approach by completely scrutinizing 11 other taxonomic works and putting the potential taxa contained in these works into explicit relationship with their own checklist taxon concepts. Table 1 gives an overview of the basic data contained in this work.

Table 2: Analysis of the "traditional" synonymy for the 1548 accepted names (taxa) in KOPERSKI & al.

Relationship	No. of taxa	% of all taxa
Cite a basionym	803	52
Cite other homotypic synonym(s)	470	30
Cite heterotypic synonym(s)	612	40
Cite misapplied name(s)	48	3

We can deduce concept instability on the higher rank level and nomenclatural instability from the analysis of the "traditional" synonymy for the 1548 taxa listed (Table 2). For example, the presence of a basionym normally indicates a past change in the genus-level classification of the species or a change in the rank of a species or infraspecific taxon. Similarly, homotypic synonyms mostly indicate a return from such a change towards the original classification. Altogether, for 948 taxa (61 %) at least one such name change has occurred in the past.

The citing of heterotypic synonyms is less easily interpreted. It may indicate that different taxonomic opinions exist with respect of the circumscription of the taxon, and that KOPERSKI & al. favour a wider view (“lumpers” vs. “splitters”). Heterotypic synonyms also imply a relationship of the underlying concepts – at least for the type specimen of the synonym, the concepts are overlapping. However, a list of heterotypic synonyms may also show the productivity of some past author in generating infraspecific taxa based on characters which did not pass the test of time. It may be interesting to note, however, that of the 600 taxa (39%) with stable names, another 214 (another 14 % of all) cite heterotypic synonyms, indicating that at least some of their data may have been referred to by a different name.

These are the kind of data many monographs and checklists can deliver. If we now turn our attention to the cited concept relationships between the potential taxa in the other scrutinised references and the concepts lying behind the accepted names in the “Reference list of German mosses” we can draw some conclusions with respect to concept stability.

The approach taken here is to handle potential taxa as sets of elements and use simple relationships from set theory (see Table 3) to describe their interaction. Oriented relationships between two sets PT1 and PT2 could also be treated quantitatively with a pair of values (α,β) where α and $\beta \in [0,1]$ and where α describes the fraction of PT1 that belongs to PT2, and β describes the fraction of PT2 that belongs to PT1. This approach looks attractive, but we lack the criteria to fix that fraction, and consequently we have not addressed the consequences such quantification would have for the calculations done by the transmission engine (see below).

Table 3: Basic concept relationships

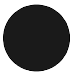
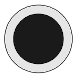
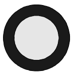
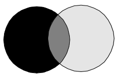
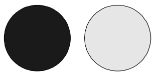
Basic relationship	Representation
R1: congruent PT_1 and PT_2 are congruent $PT_1 \equiv PT_2$ $x \in PT_1 \Leftrightarrow x \in PT_2$	
R2: included in PT_1 is included in PT_2 $PT_1 \subset C_2$ $x \in PT_1 \Rightarrow x \in PT_2, \exists y \in PT_2 \mid y \notin PT_1$	
R3: includes PT_1 includes PT_2 $PT_1 \supset PT_2$ $x \in PT_2 \Rightarrow x \in PT_1, \exists y \in PT_1 \mid y \notin PT_2$	
R4: overlaps PT_1 and PT_2 overlap each other $PT_1 \oplus PT_2$ $\exists x \in PT_1 \mid x \notin PT_2, \exists y \in PT_2 \mid y \notin PT_1,$ $\exists z \in PT_1 \mid z \in PT_2$	
R5: excludes PT_1 and PT_2 exclude each other $PT_1 \,!\, PT_2$ $x \in PT_1 \Rightarrow x \notin PT_2$	

Table 3 lists the basic (oriented) relationships from set theory that are fundamental for the description of the connection between two taxonomic concepts PT_1 and PT_2 (Representation after KOPERSKI & al., 2000).

Note that for any given PT_i and PT_j there is exactly one such relationship between PT_i and PT_j (even if this relationship might be unknown).

Table 4 asserts that in 97% of cases the "Reference list of German mosses" agrees with at least one other reference on the concept. This is reassuring - it shows that the checklist is based on past revisionary work and does not deviate radically from existing treatments. The data are also reassuring in that most of the non-congruent relationships depict inclusion (one way or the other), and not overlap, the latter being more complicated to handle (see GEOFFROY & GÜNTSCH, 2003).

Table 4: Concept synonymy for the 1548 accepted names (taxa) in KOPERSKI & al. (2000)

Concept synonymy	No. of taxa	% of all taxa
Cite congruent concept(s)	1.509	97
Cite „wider" concept(s)	515	33
Cite "narrower" concept(s)	267	17
Cite overlapping concept(s)	90	6
Cite "disjoint" concept(s)	11	1

In order to get a more precise picture of concept stability, Table 5 combines information from the "concept" synonymy with that from "traditional" synonymy.

Table 5: Analysis of the concept stability for the 1548 accepted names (taxa) in KOPERSKI & al. (2000)

Stability	No. of taxa	% of all taxa
Show stability (cite at most homotypic synonyms and no concepts except congruent concepts)	550	35
Show possible instability (cite heterotypic synonyms or misapplied names but no concepts except congruent concepts)	310	20
Show explicit instability (cite other concepts than congruent concepts)	688	45

This leads to the somewhat depressing conclusion that for at least 45% of the taxa there have been changes in the concept over time.

How far-reaching are these changes in concepts? A measure of instability of concepts would be useful to indicate the size of the problem when bringing together information gathered from different sources. We are working on a general approach using the statistical distribution of the concepts cited over the five relationship categories.

Such a measure could be

$$\mu = \frac{\sigma}{\bar{x} \cdot \sqrt{N-1}}$$

where N is the number of all cited concepts, $\bar{x}$ is the average of cited concepts for each relationship category and σ is the corresponding standard deviation. This "normalised measure" makes distributions of cited concepts for accepted names comparable even if N varies for each of them.

However, a purely statistical approach accounts neither for weighting of expert opinion nor for trends in time. As a consequence, for the present analysis we used only the number of cited relationship categories other than congruency (Table 6), although we clearly recognise the shortcomings of this approach. As we can see, in 158 cases (10% of the studied taxa) at least 3 different concepts are involved (and this using only 12 different sources, of which some apply only to a subset of the taxa!).

Table 6: Instability classes for the 688 taxa which showed concept instability. Only the minumm number of concepts involved can be assessed, because relationships are only known with respect to the accepted concept (sec. KOPERSKI & al.), not among the older concepts.

Instability class	No. of taxa	% of all taxa
Cite one relationship other than congruency (minimum of 2 concepts involved)	530	34
Cite two relationships other than congruency (minimum of 3 concepts)	122	8
Cite three relationships other than congruency (minimum of 4 concepts)	35	2
Cite all four relationships other than congruency (minimum of 5 concepts)	1	0

This is aggravated by the fact that among the 550 (35%) stable taxa in Table 5, only 207 have always been known under the same name. Figure 1 depicts the resulting situation. In terms of databasing this means that only for 13% of the taxa the name can serve as a direct index to other data (and this can only be said for the set of treatments scrutinised by the authors).

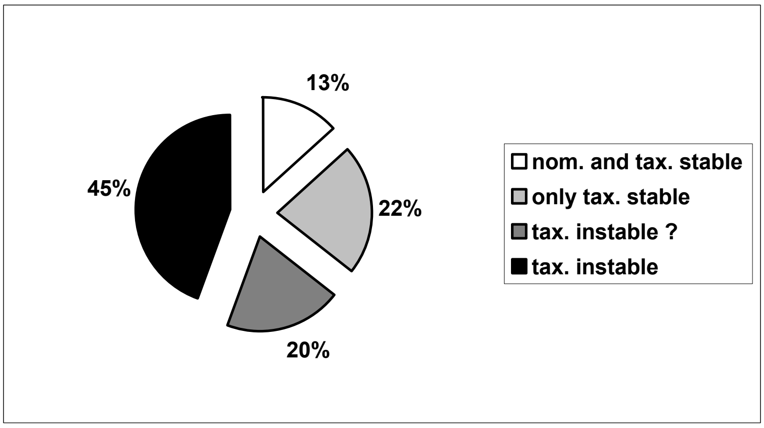


Figure 1: Nomenclatural (nom.) and taxonomic (tax.) stability

This clearly demonstrates that the answer to the initial question is no: names alone cannot serve as reliable keys for biological information, because “taxonomic database systems that use names as indexes to taxa are unable to distinguish between two concepts of the same name” and “cross-classification comparisons of taxonomic concepts are essentially impossible when taxon names and not concepts are the basis of a taxonomic database system" (both quotes from Beach & al., 1993).

The problem to be solved

Mankind's accumulated biological knowledge base is widely distributed, and we presume that the information age will not change this basic situation. In most current sources the information is linked to a (scientific) taxon name only, without indication of the concept behind the name. With the new information technologies and the World Wide Web harvesting and merging information from multiple sources, a stable indexing system is becoming crucial for further scientific research in biology. It is not fortuitous that the biological community is devoting increasing resources towards the creation of some kind of universal (taxonomic) name server, e.g. in the context of the "Species 2000" project (BISBY, 2002) and its partner databases, or in the ECAT program of the Global Biodiversity Information Facility (GBIF, 2002).

Strictly speaking, every particular use of a scientific taxon name could imply a specific concept of the taxon. As demonstrated above, this (implicit and/or explicit) concept differs often enough from the concepts implied in other sources. Factual information attached to a concept can of course be "transmitted" without any loss of accuracy to any other concept, if the two concepts circumscribe the same taxon, i.e. if they are congruent. If differing concepts are involved the indiscriminate transmission of the information could lead to wrong conclusions. Therefore it is necessary to study the implications of the relationship between potential taxa on the transmission of information between them (Figure 2). The final users of the gathered information need to get notice about possible caveats caused by the transmission process, so that they will be able to exploit the results correctly.

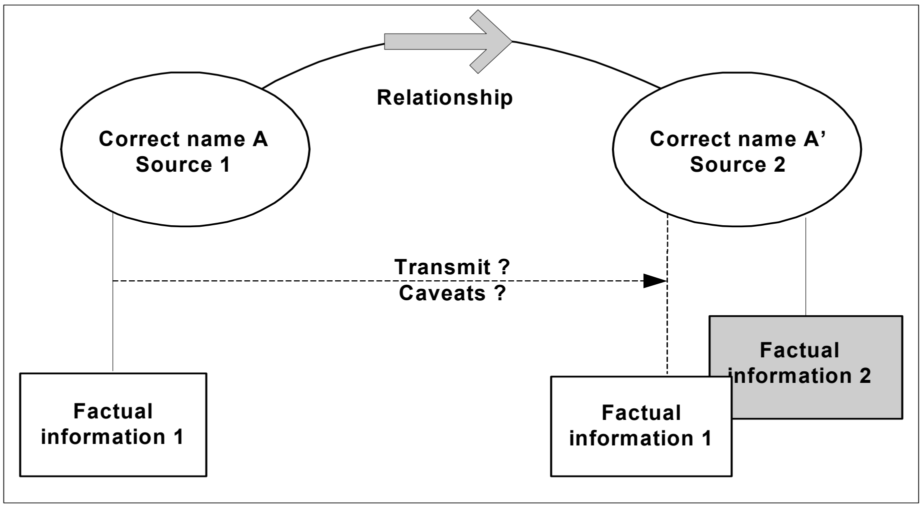


Figure 2: Are we allowed to transmit factual information?

From an overall point of view, factual information stemming from different data providers (most of them actually data bases – the factual data bases) should therefore not be transmitted to end-users without being previously processed by a "transmission engine".

Devising an architecture of the system

The MoReTax project was dedicated to address particularly the structure and the elements of such a transmission engine at a theoretical level (BERENDSOHN & GEOFFROY, 2001). Figure 3 depicts the envisioned information flow mediated by the engine. The project concluded that the basic components of such a system are:

- A network of relationships between concepts, the "potential taxon graph"(GEOFFROY & GÜNTSCH, 2003).
- Rules to calculate relationships between any two "potential taxa" (GEOFFROY & GÜNTSCH, 2003) and rules to handle the problem of factual information applicability (GEOFFROY & BERENDSOHN, 2003), both of which influence the transmission of factual information within the "potential taxon graph"
- A set of parameters with which rules can be adjusted and the transmission of factual information to the end-user interface can be controlled ("rule tuning", GEOFFROY, 2003).

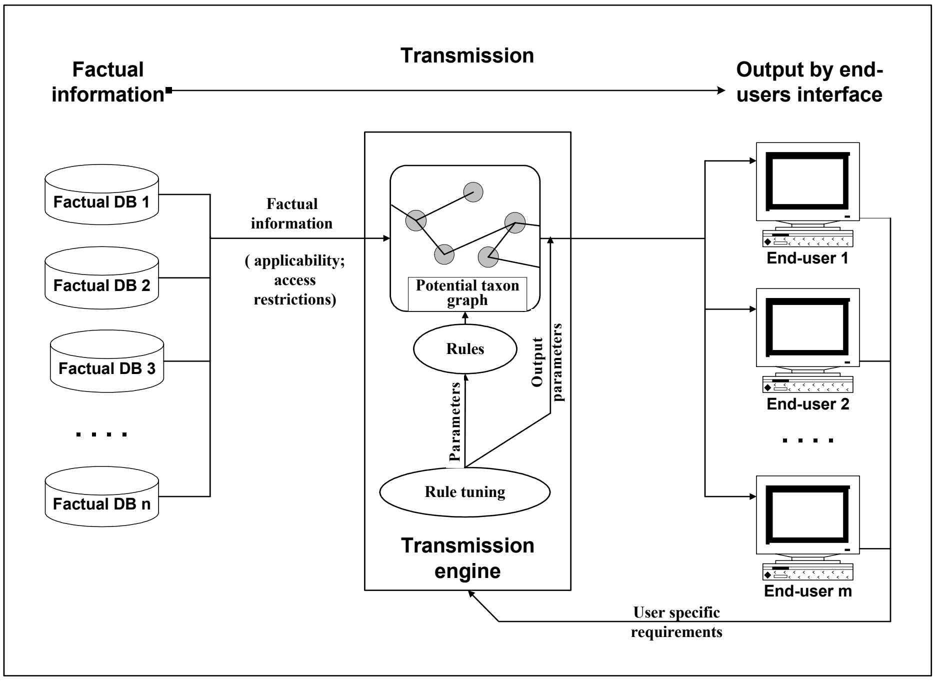


Figure 3: The information flow through the transmission engine

Furthermore, the data providers may impose requirements for the use of their data. Such access restrictions form inputs for the transmission engine and influence output for different user groups (setting of output parameters).

This in turn makes it necessary to classify users into groups based on access permissions. In addition, user-sided output should be influenced by the degree of expertise and the kind of information needs the user expresses (user requests launch the transmission engine). This part of the system cannot be solved in theory, it will form the core set of problems to be solved in the implementation of the system. Solving the "transmission" problem sums up to nothing else as to describe in detail how the transmission engine could work.

If we consider the different components and actors involved in setting up, maintaining and exploiting this information system we get an architecture centred on a core database, which in essence stores potential taxa and the relationships between them (see BERENDSOHN & al., 2003). Rules (e.g. as stored procedures) and tuning parameters can also be stored in the core database. Other important system components include:

- Factual databases, which are the sources of the potential taxa to be stored in the core database, and which also provide the factual information linked to them. In case a data provider cannot be accessed on-line, the factual information has to be stored in the core database, too.
- A "taxonomic editor", which enables experts to add or edit the taxonomic or nomenclatural data and assign relationships between potential taxa (GÜNTSCH & al., 2003).
- A "rule tuner", which enables system managers to undertake rule adjustments by setting parameters (GEOFFROY, 2003).
- End-users (people or systems) who query the information system and get the factual information transmitted by the "transmission engine".

Figure 4 sketches the architecture of such a system.

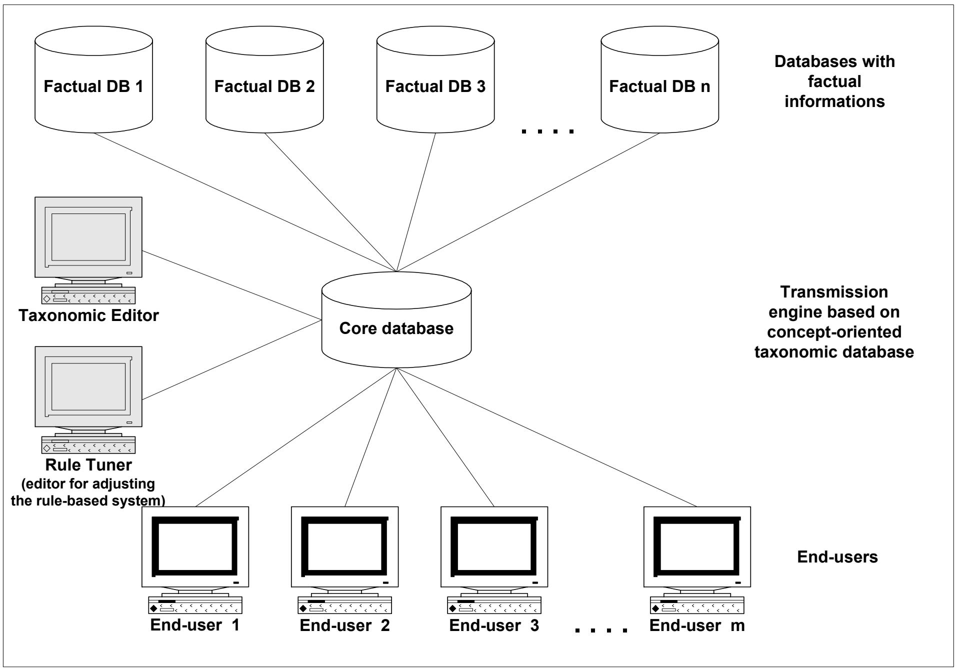


Figure4: Architecture of the system

The communication between the factual databases and the concept-based taxonomic core occurs in two different processes. A connected factual database has to provide and maintain basic metadata in the core system, i.e. the core system has to either import or otherwise be aware of taxonomic information (that means scientific names, authors, references, taxonomies, synonyms etc.) in the factual database. The second process is the actual retrieval of factual information from the linked databases, steered by the transmission engine in the core, which determines appropriate sources for the user query through the application of rules to the potential taxon graph.

For each of the last three system components user-interfaces must be created in order to allow the edition of taxonomic data and of concept relationships between potential taxa ("taxonomic editor"), to adjust the rules by setting parameter values ("rule tuner"), and to handle queries from end-users as well as the corresponding results.

In order to achieve a flexible decentralised use of the (possibly distributed) information system, remote tools and interfaces should be implemented for the communication between the different users ("clients") as well as the factual databases and the core database ("server"). The implementation of a remote tool for a centralised taxonomic database system can be based on either of the following principles:

- The client program consists of software specifically designed and implemented for the intended task.
- The client is realised with software (typically a World Wide Web browser), which is installed on the majority of computers anyway. With this approach, all data and the entire set of taxonomic rules are located at the server's side and forms are dynamically created.

In general, the second option is considered preferable because the client software will run on any operating system and users will not have to install special software other than a browser. Over the past years upward compatibility of browsers has always been given, so that system updates due to new operating systems etc. would only affect the server software. However, care has to be taken to closely adhere to common WWW standards on the server side. On the other hand, an implementation of specific parts of the system in the form of web services may lead to site specific tools being developed to work on special facets of the system. The development of a large-scale co-operative approach is indispensible for the development of a functional system, so care should be taken to allow for the integration of a wide diversity of implementation approaches.

References cited

ANONYMOUS (2002 [27 Dec]): VegBank Taxonomic Data Models. Ecological Society of America. http://vegbank.nceas.ucsb.edu/vegbank/design/planttaxaoverview.html

BEACH, J. H., PRAMANIK, S. & BEAMAN, J. H. (1993): Hierarchic taxonomic databases. Ch. 15 (pp. 241-256) in: FORTUNER, R. (ed.): Advances in computer methods for systematic biology: artificial intelligence, databases, computer vision. John Hopkings University Press, Baltimore.

BERENDSOHN, W. G. (1995): The concept of "potential taxa" in databases. Taxon 44: 207-212.

BERENDSOHN, W. G. (1997): A taxonomic information model for botanical databases: the IOPI model. Taxon 46: 283-309.

BERENDSOHN, W. G. & GEOFFROY, M. (2001 [27 Dec 2002]): MoReTax - Modelling of rule-based functions for a computerised system to link complex taxonomic concepts with factual data of relevance to nature conservartion. http://www.bgbm.org/BioDivInf/Projects/MoReTax/.

BERENDSOHN, W. G., DÖRING, M., GEOFFROY, M., GLÜCK, K., GÜNTSCH, A., HAHN, A., KUSBER, W.-H., LI, J.-L., RÖPERT, D. & SPECHT, F. (2003): The Berlin Taxonomic Information Model. Schriftenreihe Vegetationsk. 39: 15-42.

BISBY, F. (2002): The quiet revolution: biodiversity informatics and the Internet. Science 289:2309-2312.

FRANCKI, R.I.B., FAUQUET, C.M., KNUDSON, D.L. & BROWN, F. (1990): Classification and nomenclature of viruses. Archives of Virology Supplement 2: 1-445.

GBIF (2002 [27 Dec]): The GBIF Work Programme 2003. Global Biodiversity Information Facility Secretariat, Copenhagen. http://www.gbif.org.

GEOFFROY, M. (2003): Towards the implementation of the "Transmission Engine". Schriftenreihe Vegetationsk. 39: 87-112.

GEOFFROY, M. & BERENDSOHN, W. G. (2003): Transmission of taxon-related factual information. Schriftenreihe Vegetationsk. 39: 83-86.

GEOFFROY, M. & GÜNTSCH, A. (2003): Assembling and navigating the potential taxon graph. Schriftenreihe Vegetationsk. 39: 71-82.

GRADSTEIN, S. R., SAUER, M., BRAUN, M., KOPERSKI, M., & LUDIWIG, G. (2001): TaxLink, a program for computer-assisted documentation of different circumscriptions of biological taxa. Taxon 50:1075-1084.

GREUTER, W. MCNEILL, J., BARRIE, R., BURDET, H.-M., DEMOULIN, V., FILGUERIAS, T. S., NICOLSON, D. H., SILVA, P. C., SKOG, J. E., TREHANE, P., TURLAND, N. J., HAWKSWORTH, D. L. (editors & compilers) (2000): International Code of Botanical Nomenclature (Saint Louis Code) adopted by the Sixteenth International Botanical Congress St. Louis, Missouri, July - August 1999. Regnum Vegetabile 138. Koeltz Scientific Books, Königstein.

GÜNTSCH, A., GEOFFROY, M., DÖRING, M., GLÜCK, K., LI, J.-J., RÖPERT, D., SPECHT, F. & BERENDSOHN, W. G. (2003): The taxonomic editor. Schriftenreihe Vegetationsk. 39: 43-56.

ICZN (ed.) (1999): International Code of Zoological Nomenclature. Fourth Edition. - London: The International Trust for Zoological Nomenclature. 306 pp.

JONES A.C., SUTHERLAND I., EMBURY S.M., GRAY W.A., WHITE R.J., ROBINSON J.S., BISBY F.A. & BRANDT S.M (2000). Techniques for effective integration, maintenance and evolution of species databases. Pp 3-13 in: GÜNTHER, O. & LENZ, H.-J. (eds.): Proceedings of the 12th International Conference on Scientific and Statistical Database Management, Berlin, July 2000. IEEE Computer Society Press.

KOPERSKI, M., SAUER, M., BRAUN, W. & GRADSTEIN, S. R. (2000): Referenzliste der Moose Deutschlands. Schriftenreihe Vegetationsk. 34: 1-519.

LE RENARD, J. (2000): TAXIS, a taxonomic information system for managing large biological collections. P. 18 in: Abstracts. TDWG 2000: Digitizing Biological Collections. Taxonomic Databases Working Group, 16th Annual Meeting, Frankfurt, November 10-12, 2000.

MÖNKEMEYER, B. (1927): Die Laubmoose Europas. Andreales - Bryales. - In: RABENHORST, G. L. [Begr.]: Kryptogamenflora von Deutschland, Österreich und der Schweiz. Bd. IV. Leipzig (Geest & Portig) 960 S.

PULLAN, M. R., WATSON, M. F., KENNEDY, J. B., RAGUENAUD, C & HYAM, R. (2000): The Prometheus Taxonomic Model: a practical approach to representing multiple classifications. Taxon 49:55-75.

SNEATH, P. H. A. (ed.) (1992): International Code of Nomenclature of Bacteria, 1980 Revision. Washington.

STEARN, W. T. (1992): Botanical latin. 4th ed. - David & Charles, Newton Abbot. 546 pp.

SUTHERLAND, I., ROBINSON, J., BRANDT, S. M., JONES, A. C., EMBURY, S. M., GRAY, W. A., WHITE, R. J., BISBY, F. A.. (2000): Assisting the integration of taxonomic data: The Litchi Toolkit. Pp. 679-680 in: Sixteenth International Conference on Data Engineering.

TREHANE, P., BRICKELL, C.D., BAUM, B.R., HETTERSCHEID, W.L.A., LESLIE, A.C., MCNEILL, J., SPONGBERG, S.A. & VRUGTMAN, F. (1995): International Code of Nomenclature for Cultivated Plants. [ICNCP or Cultivated Plant Code.] Quarterjack Publishing, Wimborne.

YTOW, N., MORSE, D. R. & ROBERTS, D. MCL. (2001): Nomecurator: a nomenclatural history model to handle multiple taxonomic views. Biol. J. Linn. Soc. 73:81-98.

ZHONG, Y., JUNG, S., PRAMANIK, S. & BEAMAN, J. H. 1996: Data model and comparison and query methods for interacting classifications in a taxonomic database. Taxon 45: 223-241.

The Berlin Model: a concept-based taxonomic information model

WALTER G. BERENDSOHN, MARKUS DÖRING, MARC GEOFFROY, KARL GLÜCK,
ANTON GÜNTSCH, ANDREA HAHN, WOLF-HENNING KUSBER, JINLING LI,
DOMINIK RÖPERT & FRANK SPECHT

Taxonomic information models

The first step in the implementation of a database driven application is the definition of an appropriate information model, which has to be complex enough to meet the needs of the application and at the same time simple enough to be usable (GÜNTSCH & al., 2002). The taxonomic model has to incorporate nomenclatural rules and the traditional taxonomic relationships (synonymy, taxonomic hierarchy, etc.). In addition, it has to be capable of representing different taxonomic views in order to enable the system to express arbitrary relationships between potential taxa.

The solution presented here is based on the IOPI model (BERENDSOHN, 1997), but the process of implementation has led to several changes in the over-all design. Other concept-oriented models published over the past 6 years are cited by GEOFFROY & BERENDSOHN (2003a). The Berlin Model is addressing botanical data, but should serve for zoology as well, with some changes in the names section and the composition of nomenclatural reference citations. Because this is a physical model (i.e. the actual database design used in the implementation), the possibility of future changes to the design here presented cannot be excluded. These are and will be documented in the databased documentation attached to BERENDSOHN & al. (2002) on the WWW.

That documentation also provides links to the different projects using the Berlin Model (among others, Euro+Med, IOPI / EuroCAT, Med-Checklist, the Dendroflora of El Salvador and AlgaTerra). The core model covers nomenclatural relationships, potential taxa and their relationships, bibliographical information, and a general structure for factual data. The core model is extensible in order to meet specific project requirements by means of adding further entities and relationships. Nomenclatural type designation, for example, is a central subject of the AlgaTerra project and is thus covered in a model extension (see KUSBER et al., 2003).

For pragmatic reasons it was decided to base further specification on a relational model for the underlying database. There are clear advantages in other data models, but with the general aim of realising an implementation in the near future, the choice of using a relational model was based on the assumption that – for some time to come – relational database management systems (DBMS) will remain the standard tool for data storage. The DBMS used must be capable of processing stored procedures, functions, and triggers so that maximum integrity of taxonomic data can be achieved at database level. An MS SQL-Server 2000 database has been implemented as a documentation database, serving to store a model implementation of all core and extension tables, a reservoir for program elements related to the model (triggers, user defined functions, stored procedures), and to manage the documentation of the tables and attributes. Documentation of the core model as well as existing extensions is available on-line (BERENDSOHN & al., 2002), the list of tables and attributes being generated dynamically from the documentation database.

Methods and conventions

Because the model here described is a physical model, we refrain from using the terms "entity-type" and "entity" and refer to tables and records instead. A table is composed of a number of columns, which we call the "attributes" of the table. A value entered in a particular record for a particular attribute is entered into a field. Tables can be linked by associated attributes, the "keys", which establish a relationship between the tables. The key uniquely identifying the records in a table is the "primary key" and its attribute name is suffixed by "Id". Keys from other tables forming a relationship with a primary key are called "foreign keys" and their name carries the suffix "Fk".

In the text, table names are written in SMALL CAPS, attribute names are CapitalisedAndAgglomerated, and values are enclosed in 'single quotes'.

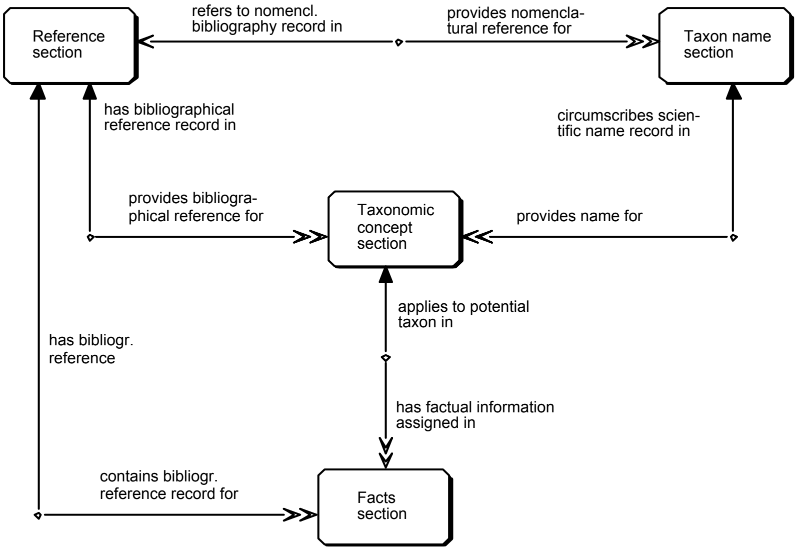

Reference section
refers to nomencl. bibliography record in
provides nomenclatural reference for
Taxon name section
has bibliographical reference record in
circumscribes scientific name record in
provides bibliographical reference for
Taxonomic concept section
provides name for
applies to potential taxon in
has bibliogr. reference
has factual information assigned in
contains bibliogr. reference record for
Facts section

Figure 1: Core functional sections of the Berlin Model, depicted as an ER-Diagram

In figures, the boxes represent tables and the connecting lines represent relationships. Figure 1, for example, depicts the relationships between the 4 basic sections of the Berlin Model's core as if they were single tables. A record in the table TAXONOMIC CONCEPTCOMPONENT always has exactly one corresponding record in the table NAMECOMPONENT, but many name records may be interpreted as taxonomic concepts. In general, the relationships are read along the connecting lines, starting with the table name, followed by the descriptive text nearest to the other table, then the "cardinality" expressed by the shape of the arrowhead pointing to the second table, and finally the name of the second table. The cardinality expresses how many records of the second table can be referred to a record of the first table. The cardinality may be 'exactly 1' (a filled-in single arrow), '0 or 1' (an open arrow), '1 to many' (a filled-in double arrow), or '0 to many' (double open arrow). The definition of cardinality here employed also states "referential integrity rules", i.e. statements guaranteeing that a foreign key always corresponds to a primary key. These rules can be declared in the database and are then enforced by the DBMS. For example, in Fig. 1, declarative referential integrity rules are used to make it impossible to delete a record of the table REFERENCE-COMPONENT while there is still a record of TAXONCONCEPTCOMPONENT or FACTS-COMPONENT referring to it. In contrast, "data integrity rules" are semantic rules for the creation, deletion, or modification of records, which have to be enforced by programmed functions or stored procedures in the database or by application programs or interfaces using the database.

The attributes of the core tables are listed in tabular form in the following text. Each attribute is listed with its short name, its type, and an explanation. In the case of foreign keys, the description is followed by an indication of the table they point to. The data types distinguished are 'int' for integer numbers, 'float' for values with decimals; for character data, 'str' (up to a fixed length, e.g. 256 characters) and 'text' (almost unlimited, cannot serve as a key); 'bool' for yes/no values; and 'date' for a complete date.

Most tables contain further attributes for technical or administrative purposes (e.g. date last changed), which are not described in detail here. Equally, the Notes attribute included in most tables is often omitted. Some tables with few attributes are described in the text. The documentation on the World Wide Web (BERENDSOHN & al., 2002) provides diagrams including all table attributes.

Basic components of the model

The core of the Berlin Model contains 24 tables, which can be grouped into four central functional sections (Figure 1): taxon names, taxonomic concepts, facts, and references. Taxon names are scientific names formed in accordance with the structural rules of the International Code of Botanical Nomenclature (ICBN, GREUTER & al., 2000), with provision made for cultivars and some more exotic constructs. Their combination with a reference, i.e. the information on who used them in what context, generates a taxonym standing for a potential taxon or taxon concept (GEOFFROY & BERENDSOHN, 2003a). An auxiliary component assembles authors into author teams for taxon names and nomenclatural references. Finally, the fact section allows the core model to be used to store modestly structured factual information. The core model and its functionality are being developed in unison by all projects. Complex factual data structures are defined in extensions to the core model and remain specific to the particular project, for example, the geographic distribution record system of the Euro+Med PlantBase project (see BERENDSOHN & al., 2002). Extensions normally refer to the Potential Taxon table. The type extension (see KUSBER & al., 2003) is an exception and may be included in the core at a later stage.

Cache fields

The Berlin Model makes extensive use of the principle of "variable atomisation" (BERENDSOHN, in press), i.e. offering the opportunity to store data in varying degrees of atomisation. The various "cache" fields are used for this purpose. Two reasons led us to the conclusion that this lack of respect for principles of normalisation and relational databases is justified. First, data imported from other systems may not have the appropriate structure and thus have to be stored in a concatenated state until they can be properly parsed into the atomised structure. Secondly, system performance is greatly enhanced by the ability to avoid join operations, being able to select from individual fields etc. Practical application of cache fields is further described in GÜNTSCH & al. (2003).

1. Nomenclature: the taxon names section

The Berlin Model attempts to separate clearly nomenclature from taxonomy, although there are some obstacles posed by the rules of nomenclature (taxon names may depend on the taxon's classification) and some grey areas where an arbitrary decision has to be taken (e.g. in the treatment of aggregates as separate names). The Berlin Model is stricter in its application of that separation than the IOPI model (BERENDSOHN, 1997), which placed some nomenclatural relationships of names within the area of potential taxa (e.g. basionym relationships). The argument was that the published information is error prone and thus should be treated as taxonomist's opinion. In the physical model here represented, we agreed to provide a "versioning" mechanism for names by archiving of all instances of a name (NAMEHISTORY), but to provide only the last version of a name as the basis for the construction of taxonyms (defined and discussed in section 3 below).

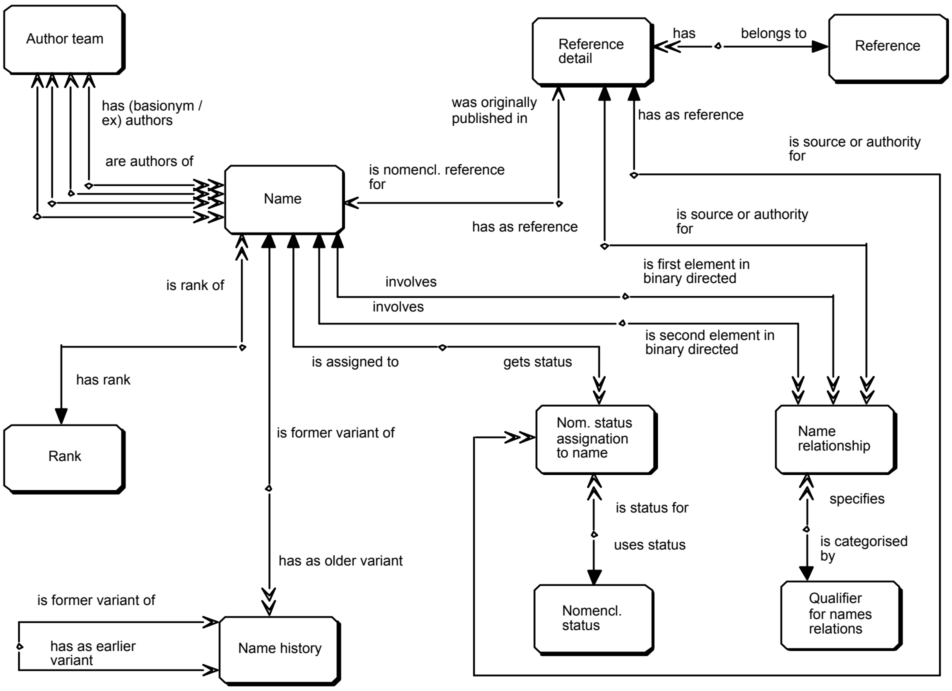


Figure 2: The names section of the Berlin Model

In the following, we will start by introducing the NAME table as the central element of the nomenclatural component (Table 1), including its relationship with the RANK table. This is followed by the description of relationships between names and the declaration of the status of a name (e.g. conservation) in accordance with the rules of botanical nomenclature (Figure 2 and Tables 1 to 3). Finally, a more detailed explanation of the NAMEHISTORY table precedes the description of the support for nomenclatural authors.

The data in the tables of the name section are used to provide a complete botanical name in the FullNameCache attribute of table NAME.

The NAME table

Every record in the NAME table represents a complete name; only the atomised author team and the rank abbreviation needed in some names are stored separately.

This represents a certain denormalisation as compared to, e.g., the implementation in PANDORA (PANKHURST, 1993) or the structures in the IOPI Model, where the name tables contain only a single attribute for a name element; the complete name is composed by means of a recurrent relationship. In PANDORA, where the distinction between names and taxa is not made, this is simultaneously representing the taxonomic hierarchy.

Name elements are stored in the attributes SupragenericName, Genus, GenusSubdivisionEpi(thet), SpeciesEpi, and InfraSpeciesEpi. The elements required for a name are determined by the rank of the name. The RankFk, a pointer to the catalogue table Rank that stores all ranks, defines the rank. The primary key RankId is an integer also indicating the position of the rank in the hierarchical taxonomic system. However, note that ranks are used here only to determine the structure of the name, not to classify them in a taxonomic sense.

For example, for all suprageneric names the only attribute used is the SupragenericName, while for all other names, the Genus field must be filled in (but not the SupragenericName). See the section on data integrity rules for further details.

The separation of the generic name (a monomial) from all other monomials is done for pragmatic reasons, as done also for the separation of the genus subdivision epithet from the species epithet (both are the first epithet in a name, and they are both parts of a binomial). Although the basic structure (monomial, binomial) of the resulting name may be similar, explicit attributes for these elements facilitate implementation, because different nomenclatural rules apply (this is even more true for zoological names).

The attribute UnnamedNamePhrase can be used to add name elements that are not in accordance with the rules of ICBN but which are needed. Examples include “unnamed taxa” (see BERENDSOHN, 1997), names from old publications (e.g., taxa with the rank “grex”), or names where intermediate ranks are cited.

Cultivars are formally named in accordance with the “Cultivated Code” (TREHANE & al., 1995) and supported by the attributes CultivarGroupName and CultivarName (= cultivar epithet).

Another issue in botanical names are hybrids (crosses between taxa). Hybrids (or graft-chimaeras) can either be named (as monomials or binomials) or they are expressed as hybrid formulas. In the latter case the names of the two parent taxa are cited with the hybrid symbol between them (the symbol for crosses is the multiplication sign, mostly replaced by an ‘x’ in databases). The HybridFormulaFlag indicates hybrid formulas. All regular name elements must be empty in such a name record and two relationships to other names are defined in the RELNAME table (see below).

This allows one not only to formulate hybrid formula like ‘*Rosa bracteata* x *Rosa carolina* ' or ‘*Populus laurifolia* x *Populus nigra’* from existing name records, it even permits one to define names for crosses between hybrids such as *‘Populus deltoides* x *P. balsamifera* x *P. angustifolia’*. Here some problems may result from the Code’s failure to distinguish primary from secondary parents in the resultant name. If such a cross is entered in the database, an arbitrary decision on the choice of the hybrid parent and the non-hybrid parent must be taken.

Named hybrids are composed as normal names, except that the hybrid symbol (or the rank-prefix “notho”) must be inserted in the concatenation of the name string. The three flags MonomHybFlag, BinomHybFlag and TrinomHybFlag indicate where this insertion should take place.

When the MonomHybridFlag is set, an x is inserted before the generic name; when the BinomHybridFlag is set, it depends on the rank: for a genus subdivision the prefix “notho” is inserted before the abbreviation of the rank, for species an x is inserted in front of the species epithet. The TrinomHybridFlag always prefixes the abbreviation of the infraspecific rank with “notho” (e.g. “nothosubsp.”).

The name elements are used to concatenate the full name in the NameCache attribute (in the case of generic subdivisions and infraspecific names with the rank abbreviation added from the rank table; in the case of hybrid formulas by combining two existing names). The contents of NameCache is combined with author teams linked from the author section of the model by means of author team foreign keys to form the full botanical name in the FullNameCache attribute. These concatenations should preferably be done directly in the database, using triggers, to avoid inconsistency between atomised and concatenated fields. However, the PreliminaryFlag offers an option to store information directly in the cache fields. This is very useful to decouple the process of importing data from the atomisation process, because it allows storage of and access to preliminary data in the database.

The attributes NomRefFk and NomRefDetailFk as a combined foreign key provide the link to the complete nomenclatural reference citation in table REFDETAIL.

Finally, the NameSourceRefFK attribute serves to store the underlying source of the names stored in the database.

Table 1: Attributes of table NAME

Short name	Type	Description
NameId	int	Primary key for table NAME
RankFk	int	Pointer to table RANK
NameCache	str	Complete Latin name string
UnnamedNamePhrase	str	Non-atomised addition to a name
FullNameCache	str	Complete Latin name string including author string
PreliminaryFlag	bool	Cache fields protected if set
SupragenericName	str	Name of taxon with rank above genus
Genus	str	Genus name or generic part of name
GenusSubdivisionEpi	str	Genus subdivision epithet
SpeciesEpi	str	Species epithet
InfraSpeciesEpi	str	Infraspecific epithet
AuthorTeamFk	int	Pointer to authors in the AUTHORTEAM table
ExAuthorTeamFk	int	Pointer to ex-authors
BasAuthorTeamFk	int	Pointer to basionym authors
ExBasAuthorTeamFk	int	Pointer to basionym ex-authors
HybridFormulaFlag	bool	If set, this name is a hybrid formula
MonomHybFlag	bool	Insert hybrid sign before monomial or genus name
BinomHybFlag	bool	Insert hybrid sign before first epithet
TrinomHybFlag	bool	Insert hybrid sign before second epithet
CultivarGroupName	str	Cultivar group designation
CultivarName	str	Cultivar epithet
NomRefFk	int	Pointer to REFDETAIL (combined foreign key), providing the nomenclatural citation
NomRefDetailFk	int	
NameSourceRefFK	int	Pointer to REFERENCE (source for current name)

Relationships between names

The table RelName (Table 2) is used to specify directed binary relationships between names. The names are represented as pointers to the Name table (first name: NameFk1, second name: NameFk2). Linking to a reference (RefFk, RefDetailFk) allows storage of the source of the relationship.

Table 2: Attributes of table RelName

Short name	Type	Description
RelNameId	int	Primary key for table RelName
NameFk1	int	Pointer to 1st name in the binary directed relationship
NameFk2	int	Pointer to 2nd name in the binary directed relationship
RelNameQualifierFk	int	Pointer to RelNameQualifier table providing the relationship category
RefFk	int	Pointer to RefDetail (combined foreign key), indicating the source reference for the relationship
RefDetailFk	int	

The relationships are classified and described in the table RelNameQualifier, which only contains its primary key and the attribute RelNameQualifier. The latter describes the directed relationship between the first and the second name. RelNameQualifier is a catalogue table, so new relationships (e.g., for different forms of orthographic variants) can be defined at any time. Currently, the following values for RelNameQualifier are defined:

- For nomenclatural relationships: 'is basionym for'; 'is later homonym of'; 'is replaced synonym for'; 'is validation of'; 'is later validation of'; 'is type of'; 'is conserved type of'; 'is rejected type of'; and (under discussion:) 'is conserved against'; 'is rejected in favour of'.
- For hybrid formulas: 'is first parent of''; 'is second parent of'; 'is female parent of'; 'is male parent of'.

The possibility of including an additional attribute to describe the implicit inverse relationship was discussed. In some cases this is perfectly possible, but in other cases it may lead to inexact statements (e.g. in the case of rejection or conservation ruled by the Code) or to truisms (has second parent, has basionym, etc.). We therefore decided to state only the explicitly defined relationship.

We should point out once more that this part of the model handles only nomenclatural relationships. These are those relationships which are used to compose the name (e.g. hybrid formulas), which form the base for the composition of the name (e.g. basionyms or names serving as types of taxa above the rank of species), or which are ruled by the International Botanical Congress and laid down in the Code itself (such as conservation). These relationships do not depend on a particular concept of the taxon. All relationships resulting from a taxonomic opinion (even if it is absolutely uncontroversial), such as synonymies of any kind, are handled in the context of potential taxa and taxonyms (in section 3 below).

Nomenclatural status

A set of nomenclatural categories exists that describe the adherence to the rules of the Code (or failure to do so) in short form. For example, a name published on or after

January 1, 1953 without a clear indication of the rank of the taxon concerned is not validly published, it becomes a ‘nomen invalidum’ or ‘nom. inval.’ A name found in literature, which apparently has never been described fully, is a ‘nomen nudum’ or ‘nom. nud.’ A name that had already been published as a botanical name with a different type, is a later homonym and a ‘nomen illegitimum’ or ‘nom. illeg.’ The catalogue table NomStatus contains such status categories (attribute NomStatus).

Note that there exists an overlap with nomenclatural relations. For example, a ‘nomen conservandum’ is often cited in publications (as ‘nom. cons.’) without giving the related name it is conserved against. It may thus exist as a status and not as a relation. In turn, if such a relation has been defined, a data integrity rule should perhaps enforce coherence between the two partial systems.

Addenda such as “comb. nov.” and “nom. nov.” that are found in literature and which are sometimes transcribed into databases by inexperienced data entry personnel do not belong here, since they are meaningful only in the context of the original publication.

Several status assignations may refer to the same name, and several sources may make the same assignation, so the status is linked to a name by means of the table NomStatusRel (Table 3), where the source reference is stated. A triple primary key consisting of NameFk, NomStatusRefFk, and NomStatusFk allows the handling of multiple references that assign the same status. The DoubtfulFlag is set when the source cites but questions the attribution of a status (e.g. ‘nom. illeg.?’).

Table 3: Attributes of table NomStatusRel

Short name	Type	Description
NameFk	int	Pointer to Name, part of this table's primary key
NomStatusFk	int	Pointer to NomStatus, part of this table's primary key
NomStatusRefFk	int	Pointer to RefDetail (combined foreign key), indicating the source reference for the status assignation
NomStatusRef-DetailFk	int	
DoubtfulFlag	bool	Flag indicating whether the nomenclatural status of this name is considered doubtful

Rank

The rank of a name is a very important structural component because it determines the structure of a name and because it may have to be included as a component of the name. The catalogue table Rank contains all known and currently acceptable taxonomic ranks of nomenclatural standing - it excludes symbols only used in historic treatments as well as some “ranks” historically used for hybrids (e.g. “grex”) or cultivars (“cv.”).

The values ‘aggregate’ and ‘species group’ have been included as ranks in spite of the fact that they do not have a nomenclatural standing. However, their (correct) placement as taxonomic concepts would have been impractical, because (i) they are cited differently from their base names (normally without authors, but with the addendum ‘species group’ or ‘aggr.’), and (ii) because they are normally defined in the same publication as the included microspecies carrying the same name - and that would violate the rule that only one concept of a taxon is defined in a single reference. All workarounds would have been cumbersome and would have affected the processing of all other names, so these constructs are treated pragmatically, as independent names without authors and with their own rank (just above species).

The RankId serves as the primary key of the table and at the same time it indicates the position of a given rank in the hierarchy. This greatly facilitates the implementation of many rank-dependent data integrity rules (see that section below). For subdivisions of genera and for infraspecific names the values in the attribute RankAbbrev (e.g. 'subgen.' or 'subsp.') are used in the concatenation of the full botanical name in the NameCache and FullNameCache fields of the table NAME (see above).

Name history

The NAMEHISTORY table serves as an archive of updates of the NAME table in the system (not for historical records on nomenclatural activities). The recursive foreign key SuccNameHistId in NAMEHISTORY allows the reconstruction of past updates of the NAME table in the history of the system. These events can be dated by means of the Created_When attribute. A new record is added to the NAMEHISTORY table whenever an existing record in the NAME table is updated. The original data are copied to the corresponding fields in NAMEHISTORY, but (with the exception of the RankFk) all data defined by foreign keys in NAME (e.g. author teams, references) are concatenated and archived as text.

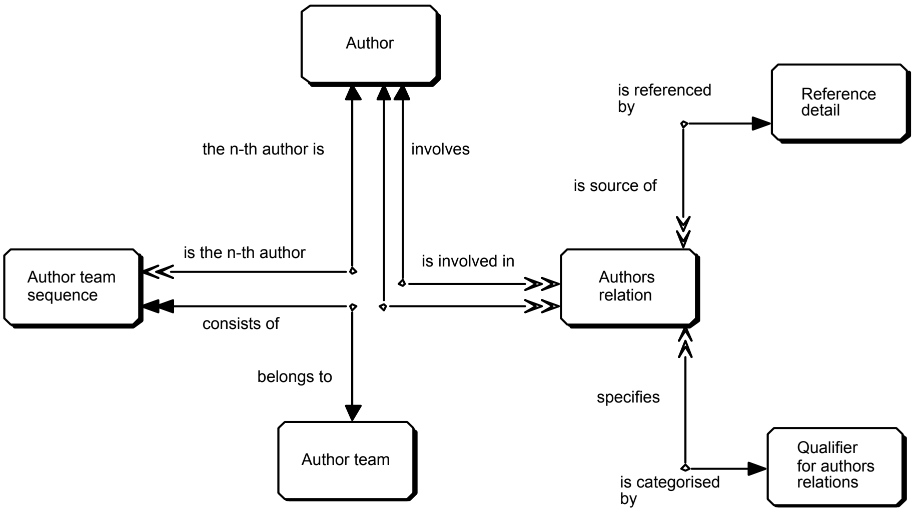


Figure 3: Context of authors and author teams

Name authors

In botanical name citations, the nomenclatural authors traditionally play an important role as a recommended part of the name's citation. The present model (Figure 3) follows to a large extent the one developed by ELANKOVAN & al. (1996). Authors' names are normally abbreviated, and often more than one author is present (author team). The table AUTHORTEAM (Table 4) thus forms the central interface of the author module (a single author is considered a team with just one member). Its primary key (AuthorTeamId) is referenced four times from the NAME table because both basionym and combination authors may attribute authorship to others ("ex-authors"). The concatenated team of authors as a string of author abbreviations is stored in the AuthorTeamCache. Similar to other cache fields, its content can either be composed by a trigger from the individual authors forming the team, or the field can be used to capture preliminary author strings (in which case the PreliminaryFlag must be set to true to protect the cache field against overwriting by the trigger).

Table 4: Attributes of table AUTHORTEAM

Short name	Type	Description
AuthorTeamId	int	Primary key for table AUTHORTEAM
AuthorTeamCache	str	Complete author team string
PreliminaryFlag	bool	Author team cache protected

Individual authors are described in the table AUTHOR (Table 5). The attribute Abbrev holds a standard abbreviation of the author's name, which is used in the nomenclatural citations. The reference to the standard used (e. g. BRUMMITT & POWELL, 1992) is given in the attribute NomStandard. Further biographic details of the author may aid in the identification of the correct author. The AreaOfInterest denotes the author's specialisation using the abbreviations defined by BRUMMITT & POWELL (1992) (S,M,A,P,B,F,L,C, for Spermatophytes, Mycology (fungi and lichens), Algae, Pteridophytes, Bryophytes, Fossils, Pre-Linnaean, and unspecified Cryptogamic, respectively).

Table 5: Attributes of table AUTHOR

Short name	Type	Description
AuthorId	int	Primary key for table AUTHOR
Abbrev	str	Abbreviation of author name as formally used
FirstName	str	Author's full first name
LastName	str	Author's full last name
Dates	str	String indicating the author's lifespan
AreaOfInterest	str	Research area in which the author is specialised
NomStandard	str	The abbreviation standard for nomenclatural authors

The AUTHORTEAMSEQUENCE table has only three attributes. It resolves the many-to-many relationship between AUTHOR and AUTHORTEAM (attributes AuthorTeamFk and AuthorFk) and at the same time it provides the correct order of the authors in the team. The Sequence attribute defines the correct position for the author in the team.

Authors' names may be abbreviated according to different standards and several orthographic variants of their names may exist in the database (for example through transliterations: Smirnow or Smirnov or Smirnoff). Since the name section of the Berlin Model aims at establishing a single nomenclatural reference for a given project, a single standard abbreviation for a person is aimed at, too. For that purpose, the RELAUTHOR table allows the establishment of a binary relationship, linking a variant or an old standard record with the accepted one. The relationship is qualified by the means of the RELAUTHORQUALIFIER, which, beside its primary key, has only the attribute

RelAuthorQualifier. Values include ‘orthographic variant’, ‘old standard for’ and ‘equals’ - the latter for the case where an author has changed name e.g. by marriage.

Table 6: Attributes of table RELAUTHOR

Short name	Type	Description
RelAuthorId	int	Primary key for table RELAUTHOR
AuthorFk1	int	Pointer to AUTHOR (1st author)
AuthorFk2	int	Pointer to AUTHOR (2nd author)
RelAuthorQualifierFk	int	Pointer to RELAUTHORQUALIFIER
RefFk	int	Pointer to REFDETAIL (combined foreign key), indicating the source reference for the relationship
RefDetailFk	int	

Data integrity rules for the name section of the model

(Note that many rules with respect to obligatory fields only apply if the PreliminaryFlag is not set.)

Name elements

- attributes SupragenericName, Genus, GenusSubdivisionEpi, SpeciesEpi, and InfraSpeciesEpi consist of a single word without spaces (hyphens allowed within the word)
- SupragenericName and Genus are capitalised, epithets do not contain capital letters
- attributes CultivarName and CultivarGroupName may consist of letters, hyphens, and spaces
- single quotation marks used to denote cultivar names are not included in the field CultivarName (= cultivar epithet)
- brackets or parentheses used to denote the CultivarGroupName are not to be included in the field

Rank

- it is not possible to change the rank of a name, except by creating a new name and a potential taxon name referring to it

• If the rank is above genus

- the attribute SupragenericName in NAME must be filled
- the attributes Genus, GenusSubdivisionEpi, SpeciesEpi, InfraSpeciesEpi, CultivarName, and CultivarGroupName in NAME must be empty
- none of the 4 hybrid flags in NAME may be set
- no basionym relationship involving the name may be defined in RELNAME

• If the rank is genus

- the attribute Genus in NAME must be filled
- the attributes SupragenericName, GenusSubdivisionEpi, SpeciesEpi, and InfraSpeciesEpi in NAME must be empty

• If the rank is below genus and above aggregate and species group

- the attributes Genus and GenusSubdivisionEpi in NAME must be filled
- the attributes SupragenericName, SpeciesEpi, and InfraSpeciesEpi in NAME must be empty
- the abbreviation of the rank must be inserted in between genus and generic subdivision epithet to form the cached name

• If the rank is aggregate or species group

- the attributes Genus and SpeciesEpi in the NAME table must be filled
- the attributes SupragenericName, GenusSubdivisionEpi, and InfraSpeciesEpi in NAME must be empty
- the primary key of the name may not appear in RELNAME or NOMSTATUSREL
- no author teams may be assigned
- for the cached name, ‘aggr.’ or ‘species group’ is added to the name

• If the rank is species

- the attributes Genus and SpeciesEpi in NAME must be filled
- the attributes SupragenericName, GenusSubdivisionEpi, and InfraSpeciesEpi in NAME must be empty

- the primary key of the name may not appear as NameFk2 in RELNAME for 'is type of' relationships (types of names in the rank of species or below are not names)

- If the rank is below species
- the attributes Genus, SpeciesEpi, and InfraSpeciesEpi in NAME must be filled
- the attributes SupragenericName and GenusSubdivisionEpi in NAME must be empty
- the abbreviation of the rank must be inserted between species epithet and infraspecific epithet to form the cached name
- if species epithet and infraspecific epithet are the same (autonym), then the author team string is inserted after the species epithet instead of after the entire name
- the primary key of the name may not appear as NameFk2 in RELNAME for 'is type of' relationships (types of names in the rank of species or below are not names)
- the nomenclatural author citation for autonyms is stored as if it were the author citation of the name; it is put in its right place (before the abbreviation of the rank and the second epithet) upon output
- no nomenclatural reference is assigned to an autonym

Hybrids

- if the HybridFormulaFlag is set in NAME, the attributes SupragenericName, Genus, GenusSubdivisionEpi, SpeciesEpi, InfraSpeciesEpi, CultivarName, and CultivarGroupName in NAME must be empty, and the other three hybrid flags may not be set
- formulas may consist only of names of generic rank or below
- the rank of the record of a hybrid formula in NAME as indicated by RankFk is the lowest rank among its parents
- in RELNAME, hybrid parents may only be defined for those records in NAME which have at least one of the four hybrid flags set
- hybrid formulas cannot have other than parent relationships defined in RELNAME (at least for the values presently defined)
- a graft chimaera cannot be the base name of cultivars, cultivar groups or unnamed taxa, nor the parent of a hybrid
- a hybrid formula cannot have an assigned author(team)

Unnamed taxa

- are accepted only at a rank below genus; the name phrase must therefore always be combined with at least a generic name

Name relations

- the creation or deletion of a nomenclatural relationship between names that can be expressed by a value in NOMSTATUS may trigger a corresponding change in NOMSTATUSREL
- a name can have only a single basionym OR a single replaced synonym
- records in RELNAME must not result in circular references

Nomenclatural status

- a later homonym in RELNAME should also have a nomenclatural status of 'nom. illeg.'
- a name conserved against another in RELNAME should also have the status 'nom. cons.'
- a name rejected against another in RELNAME should also have the status 'nom. rej.'

Authors and author teams

- any author may belong to a given team only once
- any author team consisting of the same authors in the same sequence is the same team and must occur not more than once
- a BasAuthorTeamFk can only be assigned if an AuthorTeamFk is present
- an ExAuthorTeamFk or an ExBasAuthorTeamFk can only be assigned if the respective author team is assigned as well
- records in RELAUTHOR must not result in circular references

2. Bibliography: the reference section

Nomenclatural and bibliographic citations

References were not treated in the IOPI model as published in BERENDSOHN (1997), but an extensive analysis had been included in an earlier version (BERENDSOHN, 1994). The reference section of the Berlin model (Figure 4) holds structured information for two different kinds of citations. Nomenclatural reference citations are used only by the NAME section (NomRefFk and NomRefDetailFk in table NAME), while bibliographical reference citations are linked with various tables throughout the model. The latter are used to reference factual data, to record sources of relationships (e.g. in RELPTAXON, RELNAME), and they are part of the potential taxon primary key where they denote the circumscription reference of the taxonym (see definition of this term in Geoffroy & Berendsohn, 2003a). A single reference record can be used to hold either or both kinds of references.

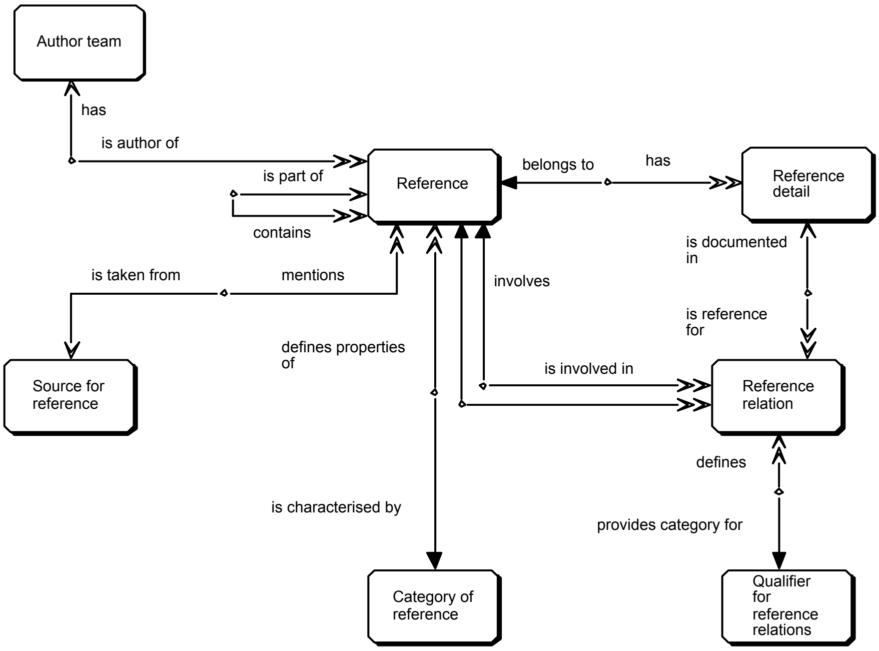


Figure 4: The reference section of the model

The separation of nomenclatural and bibliographic citations is difficult. On the one hand, both refer to the same object, a publication. On the other hand, taxonomists traditionally abbreviate nomenclatural citations. Unfortunately, no clear rules exist as to where to abbreviate and how far, although certain standards (BPH, LAWRENCE & al., 1968; TL-2, STAFLEU & COWAN, 1976-1988, both with later supplements) are widely used. Botanists largely agree to abbreviate the names of the actual authors of the name or the combination (see section on name authors above). However, opinions are split with regard to the citation of "in" authors, which are part of a bibliographic citation rather than part of a nomenclatural citation. For example, the recently published Flora of Nicaragua (STEVENS & al., 2001) uses abbreviations for both kinds of authors: *"Polygala gracilis* Kunth in Humb., Bonpl. & Kunth, Nov. Gen. Sp. 5: 401. 1823." In contrast, KUSBER & al. (2003)cite the "in" authors by their full name: "*Scenedesmus quadricauda* (Turpin) Bréb. in Brébisson, L. A. & Godey, L. L.: Alg. Falaise: p. 66. 1835." The latter is also the procedure followed (not ruled!) by the editors of the Botanical Code (GREUTER & al., 2000). The model accommodates both (and other) options for output, but the presence of calculated fields (the cache fields) makes it necessary to

Table 7: Attributes of table REFERENCE

Short name	Type	Description
RefId	int	Primary key for table REFERENCE
RefCache	str	Full bibliographic citation including data referenced by InRefFk
NomRefCache	str	Full nomenclatural citation including data referenced by InRefFk
PreliminaryFlag	bool	When set, the content of RefCache is protected
RefCategoryFk	int	Pointer to REFCATEGORY specifying the nature (and thus structure) of a reference
InRefFk	int	Recursive pointer for hierarchical relations ("in")
Title	str	Bibliographic title
NomTitleAbbrev	str	Nomenclatural abbreviation of title
NomAuthorTeamFk	int	Pointer to AUTHORTEAM for nomencl. references
RefAuthorString	str	Bibliographical author(team)
Edition	str	Edition
Volume	str	Volume and supplementary information
Series	str	Bibliographic publication series
RefYear	str	Year of publication
PageString	str	Total pages of an article or book
DateString	str	Date to denote "versions" of the same source
ISSN	str	ISSN code of the publication
ISBN	str	ISBN code of the publication
URL	str	Full URL (http:// …) for datasources on the web
ExportDate	str	Date exported (e.g. from a database)
PublicationTown	str	Place of publication
Publisher	str	Publisher
ThesisFlag	bool	Indicating if the reference is a thesis
RefDepositedAt	str	Location where reference is held (e.g. a library)
InformalRefCategory	str	Informal reference category
IsPaper	bool	Indication that the reference is printed
RefSourceFk	int	Pointer to REFSOURCE (record source)
IdInSource	str	Original source ID for imported references
NomStandard	str	The abbreviation standard for nomenclatural references which has been used

reach an agreement, preferably one that can be used generally. The triggers currently implemented for the NomRefCache use the first option (abbreviated author and title citation of the "in" reference), simply to shorten on-screen output. Thus, a bibliographical citation uses the attributes Title and RefAuthorString for the full title and author strings, whereas a nomenclatural citation uses the attributes NomTitleAbbrev and NomAuthorTeamFk for the abbreviated form.

The table REFERENCE

This table and its recursive relationship handle almost all data on references in the system (Table 7). Only information pointing to a specific part of a reference, e.g. a single page number or figure, is kept in the separate table REFDETAILS (Table 9).

Hierarchical reference structures such as articles in journals or parts of a book are accommodated by means of a recursive relationship (attribute InRefFk). This allows normalising the relationship to periodicals or books containing multiple articles. For nomenclatural citations, which are commonly abbreviated, it also helps to standardise title citations. A single recursion is very common ("in" reference, article in journal or section in book), a second one rare (e.g. part of a family treatment by an author as part of a family edited by another author and published as an article in a periodical). In theory, the model allows an unlimited number of recursions, but all implementations at the BGBM so far have agreed on limiting the number to a maximum of two, because an unknown number of recursion levels is difficult to process in triggers and other program routines. To further simplify matters, volumes of books are treated as separate records (i.e., without a link to the common book title), and information about volumes and series of periodicals is stored at article level. Further conventions agreed are to use the Volume attribute to store additional data such as supplement numbering, and to store the Series and Edition strings with their respective term or abbreviation (ed., ser., etc.).

The RefSourceFk attribute holds a foreign key to the table REFSOURCE where information about the original source of a reference record can be stored (e.g. if the record was imported). REFSOURCE contains only a text field to state the source and a Notes field, apart from its primary key.

Table 8: Values of attribute RefCategory in table REFCATEGORY

RefCategory	Description
'book'	a book
'journal'	a journal (title only)
'article in periodical'	an article referenced to a journal by means of the InRefFk
'part of other title'	other "in citations", e.g. contribution in a book
'database'	a database
'published CD'	a published compact disc or DVD
'website'	a website
'informal reference'	an informal reference
'unresolved'	a reference for which the category has not been determined yet (e.g. in imports to the cache fields)
'not applicable'	if no category can be assigned

The obligatory foreign key RefCategoryFk in table REFERENCE indicates the category of the reference record, e.g. 'book' or 'database'. The table REFCATEGORY contains only two string attributes apart from its primary key: RefCategory and RefCategoryAbbrev(iation). The category determines the structure, data integrity rules (see the section below), and rules for the concatenation of the RefCache for each reference record. The core model implements 10 basic categories (Table 8).

The table REFDETAIL

This table allows pinpointing of a place (rarely: places) within a single reference. The attribute Details holds the page number (or in some cases illustration or plate numbers, etc.). It is used by all nomenclatural reference citations, which generally state the location of the protologue.

This table holds the full nomenclatural and bibliographical citation of a reference including the details. In contrast to the bibliographical string (FullRefCache), the nomenclatural string in FullNomRefCache does not cite the author(s) of the reference, because they are already contained in the full name of the taxon (AuthorTeamFk in NAME and NomAuthorTeamFk in REFERENCE are the same).

For the current projects, we agreed to de-normalise the relationship to RefDetail (but this is not prescribed by the model). Every use of a reference by another record (every foreign key pointing to the table) creates a new REFDETAIL record. This protects the SecondarySources and other details which may be different even if the location is the same. It also prevents the establishment of several relationships to an initially empty detail record, which may later be filled in (and than be correct only for one of the relationships).

Table 9: Attributes of table REFDETAIL

Short name	Type	Description
RefDetailId	int	First part of primary key for table REFDETAIL
RefFk	int	Second part of primary key and pointer to table REFERENCE
FullRefCache	str	Full bibliographic citation string including details
FullNomRefCache	str	Full nomenclatural citation string including details
PreliminaryFlag	bool	Cache fields protected, if set
Details	str	Reference details, such as exact page or no. of figure
SecondarySources	str	Secondary sources (sources named in the reference)

Use of cache fields in references

There are four calculated fields in this section that provide fast access to a concatenated citation string. In table REFERENCE, RefCache holds a full bibliographical citation including the author team (RefAuthorString). NomRefCache of table REFERENCE stores the abbreviated form of this string. Nomenclatural citations have to point to a detailed part of an information source and therefore the full nomenclatural citation string is available in FullNomRefCache of table REFDETAIL. For full bibliographical references including details, attribute FullRefCache in the same table is used. A trigger or another automatic procedure can fill these cache attributes if the respective PreliminaryFlag is not set. Otherwise they provide the means to store a (preliminary) full citation string without fully atomising its content.

A thesaurus function for references

Analogous to authors' names, references may be abbreviated according to different standards and variants of titles may exist in the database. The tables RELREFERENCE (Table 10) and RELREFERENCEQUALIFIER are provided to establish binary relationships between references, e.g. to implement a thesaurus for different nomenclatural abbreviation standards. The two references in question are pointed to by the attributes ReferenceFk1 and ReferenceFk2. Setting the foreign keys RefFk and RefDetailFk can cite an optional source reference for the relationship. The foreign key RelReferenceQualifierFk determines the kind of relation. The attribute RelReferenceQualifier (the only data attribute in the table with the same name) gives the category of relationship between the two references (e.g. 'equals').

Table 10: Attributes of table RELREFERENCE

Short name	Type	Description
RelReferenceId	int	Primary key for table RELREFERENCE
ReferenceFk1	int	Pointer to 1st reference (in REFERENCE)
ReferenceFk2	int	Pointer to 2nd reference of the relationship
RelReferenceQualifierFk	int	Pointer to RELREFERENCEQUALIFIER
RefFk	int	Pointer to REFDETAIL (combined foreign key), indicating the source of the relationship
RefDetailFk	int	

Data integrity rules for the reference section of the model

Authors

- when a name is related to a reference by means of the attribute NomRefDetailFk of table NAME, the attribute AuthorTeamFk of table NAME must have the same value as the attribute NomAuthorTeamFk in the linked REFERENCE record.

RelReference

- Records in RELREFERENCE must not define circular references

Reference category

- The reference category determines the use of attributes for a given reference record. Table 11 shows all allowed attributes for the major categories. Some categories have a mandatory InRefFk attribute, which may have to point to a parental record of a certain reference category.

Table 11: Reference categories and related attributes (● = mandatory, ○ = optional, ✖ = prohibited)

Attribute	Journal (J)	Journal Article (A)	Book (B)	Part of other title (P)	Data-base (D)	Informal (I)	Unre-solved (U)
RefCache	○	○	○	○	○	○	○
NomRefCache	○	○	○	○	✖	✖	○
PreliminaryFlag	○	○	○	○	○	○	✖
RefCategoryFk	●	●	●	●	●	●	●
InRefFk, allowed parental categories	✖	J	✖	A, B, P, D, I, U	D	✖	✖
Title	○	○	○	○	○	○	○
NomTitleAbbrev	○	✖	○	✖	✖	✖	○

Table 11 (continued)

Attribute	Journal (J)	Journal Article (A)	Book (B)	Part of other title (P)	Data-base (D)	Informal (I)	Unre-solved (U)
NomAuthorTeamFk	✖	○	○	○	✖	✖	○
RefAuthorString	✖	○	○	○	○	○	○
Edition	✖	○	○	✖	○	○	○
Volume	✖	○	○	✖	✖	○	○
Series	✖	○	○	✖	✖	○	○
DateString	○	✖	○	✖	○	○	○
RefYear	✖	○	○	○	○	○	○
PageString	✖	○	○	○	✖	○	○
ISSN	○	✖	✖	✖	✖	✖	✖
ISBN	✖	✖	○	✖	✖	✖	✖
PublicationTown	○	✖	○	✖	✖	✖	○
Publisher	○	✖	○	✖	✖	✖	○
URL	○	○	○	○	○	○	○
ExportDate	○	○	○	○	○	○	○
ThesisFlag	✖	○	○	○	✖	○	○
RefDepositedAt	✖	○	○	✖	✖	○	○
InformalRefCategory	✖	✖	✖	✖	✖	○	✖
IsPaper	✖	✖	✖	○	✖	○	○
RefSourceFk	○	○	○	○	○	○	○
IdInSource	○	○	○	○	○	○	○
NomStandard	○	✖	○	✖	✖	✖	○

Examples

Examples for the placement of data elements of nomenclatural references are given in Tables 12-14.

Table 12: Attribute values for "*Oxytropis campestris* (L.) DC., Astragalogia: 74. 1802."

Table	Attribute	Record 1	Record 2
REFERENCE	RefId	1	
REFERENCE	InRefFk		
REFERENCE	RefCategoryFk	book	
REFERENCE	NomTitleAbbrev	Astragalogia	
REFERENCE	NomAuthorTeamFk	2	
REFERENCE	Edition		
REFERENCE	Volume		
REFERENCE	RefYear	1802	
REFERENCE	NomRefCache	DC., Astragalogia. 1802.	
REFDETAIL	RefDetailId	1	
REFDETAIL	RefFk	1	
REFDETAIL	Details	74	

Table 12 (continued)

Table	Attribute	Record 1	Record 2
REFDETAIL	FullNomRefCache	Astragalogia: 74. 1802.	
NAME	NameId	1	
NAME	AuthorTeamFk	2	
NAME	BasAuthorTeamFk	1	
NAME	NomRefFk	1	
NAME	NomRefDetailFk	1	
NAME	NameCache	Oxytropis campestris	
NAME	FullNameCache	Oxytropis campestris (L.) DC.	
AUTHORTEAM	AuthorTeamId	1	2
AUTHORTEAM	AuthorTeamCache	L.	DC.

Table 13: Attribute values for "*Oxytropis kotschyana* Boiss. & Hohen. in Boiss., Diagn. Pl. Orient. 9: 36. 1849."

Table	Attribute	Record 1	Record 2
REFERENCE	RefId	1	2
REFERENCE	InRefFk		1
REFERENCE	RefCategoryFk	book	part of other title
REFERENCE	NomTitleAbbrev	Diagn. Pl. Orient.	
REFERENCE	NomAuthorTeamFk	1	2
REFERENCE	Volume	9	
REFERENCE	RefYear	1849	
REFERENCE	NomRefCache	Boiss., Diagn. Pl. Orient. 9. 1849.	Boiss. & Hohen. in Boiss., Diagn. Pl. Orient. 9. 1849.
REFDETAIL	RefDetailId	1	
REFDETAIL	RefFk	2	
REFDETAIL	Details	36	
REFDETAIL	FullNomRefCache	Boiss., Diagn. Pl. Orient. 9: 36. 1849.	
NAME	NameId	1	
NAME	AuthorTeamFk	2	
NAME	NomRefFk	2	
NAME	NomRefDetailFk	1	
NAME	NameCache	Oxytropis kotschyana	
NAME	FullNameCache	Oxytropis kotschyana Boiss. & Hohen.	
AUTHORTEAM	AuthorTeamId	1	2
AUTHORTEAM	AuthorTeamCache	Boiss.	Boiss. & Hohen.

Table 14: Attribute values for *"Oxytropis prenja* (G. Beck) G. Beck in Reichenb. & Reichenb. fil., Icon. Fl. Germ. Helv. 22: 124. 1901."

Table	Attribute	Record 1	Record 2	Record 3
REFERENCE	RefId	1	2	3
REFERENCE	InRefFk		1	2
REFERENCE	RefCategoryFk	journal	article	part of other title
REFERENCE	NomTitleAbbrev	Icon. Fl. Germ. Helv.		
REFERENCE	NomAuthorTeamFk		1	2
REFERENCE	Volume		22	
REFERENCE	RefYear		1901	
REFERENCE	NomRefCache	Icon. Fl. Germ. Helv.	Reichenb. & Reichenb. fil. in Icon. Fl. Germ. Helv. 22. 1901.	G. Beck in Reichenb. & Reichenb. fil., Icon. Fl. Germ. Helv. 22. 1901.
REFDETAIL	RefDetailId	1		
REFDETAIL	RefFk	3		
REFDETAIL	Details	124		
REFDETAIL	FullNomRefCache	Reichenb. & Reichenb. fil., Icon. Fl. Germ. Helv. 22: 124. 1901.		
NAME	NameId	1		
NAME	AuthorTeamFk	2		
NAME	BasAuthorTeamFk	2		
NAME	NomRefFk	3		
NAME	NomRefDetailFk	1		
NAME	NameCache	Oxytropis prenja		
NAME	FullNameCache	Oxytropis prenja (G. Beck) G. Beck		
AUTHORTEAM	AuthorTeamId	1	2	
AUTHORTEAM	AuthorTeamCache	Reichenb. & Reichenb. fil.	G. Beck	

3. The taxonomic concept section: potential taxa and taxonyms

This section (Figure 5) provides the concept-based centre of the Berlin Model.

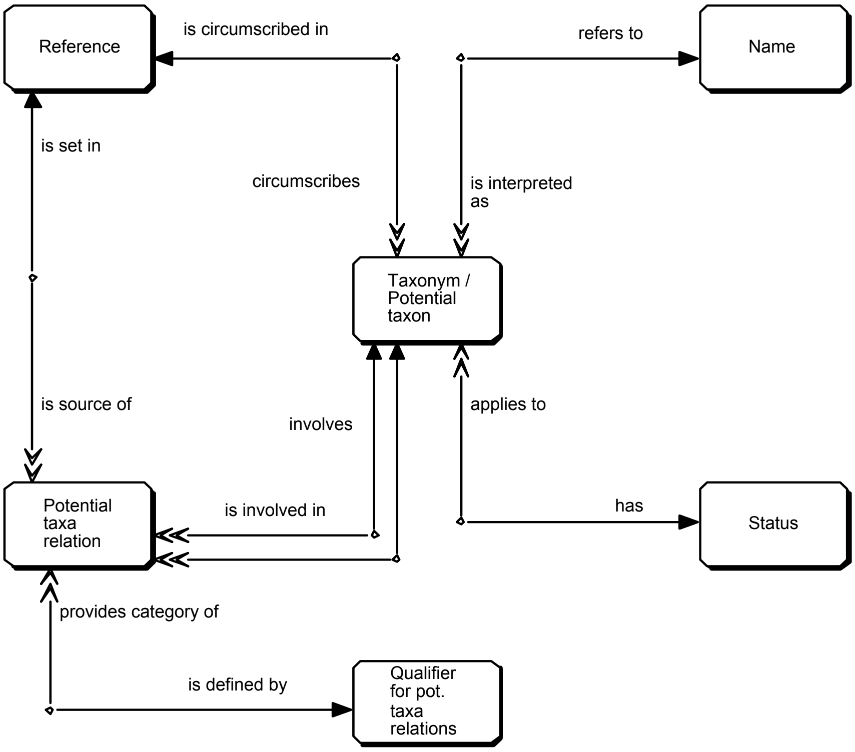


Figure 5: Concept section of the model

A taxonomic concept and its corresponding potential taxon are identified by a taxonym (GEOFFROY & BERENDSOHN, 2003a), a combination of the scientific taxon name with the bibliographic citation of the source in which it was used. Therefore the implementation of taxonomic concepts as a table (PTAXON, Table 15) is based on linking name records from the NAME table (attribute 'PTNameFk') and reference records from the REFERENCE table (attribute 'PTRefFk').

This is in contrast to other tables, where references are always indicated by means of a pointer to REFDETAIL. However, then PTAXON would have a tripartite primary key, and every query involving a potential taxon (i.e. most queries) would involve additional SQL-Join. We posit that a circumscription reference will normally not include exact page citations; citing the reference author, title, and year should suffice. Where needed, indicating exact details of the bibliographic reference for the potential taxon is still possible by means of the attribute Detail.

In the Berlin Model, the taxonomic status (accepted or correct, synonym) of a name and factual data are linked to PTAXON. Thus, we clearly differentiate nomenclatural data from all data relating to the definition and usage of taxonomic concepts.

Table 15: Attributes of the PTAXON table

Attribut	Type	Description
PTNameFk	int	Pointer to NAME and part of primary key
PTRefFk	int	Pointer to REFERENCE and part of primary key
Detail	str	Informal way to further specify the reference
IdInSource	str	ID in the original source if this taxonym was imported
StatusFk	int	Pointer to STATUS
Doubtful-Flag	int	Indicates a provisional status assignation; also serves to temporarily inactivate taxonyms in the editing process
NamePhrase	str	Additional name suffix for this taxonym (e.g. "sensu auct. ")
UseName-CacheFlag	bool	Indicates that the author string should be omitted in output; often used in combination with the NamePhrase ‘sensu...’

The attribute NamePhrase allows storing a suffix to the name, which gives an indication of the circumscription of the potential taxon but does not provide a reference (e.g. ‘sensu lato’, ‘sensu stricto’, ‘sensu auct. amer.’, ‘non Bory’, etc.). The UseNameCacheFlag makes it possible to exclude the author string from the output of the taxon name part, by enforcing the use of the NameCache instead of the FullNameCache attribute from the NAME table. This may be desired in case that the content of the NamePhrase attribute is considered to be a replacement of the author string.

Note that these attributes should not be used to replace concept relationships. For example, a non-citation which provides a reference should be treated as a taxonym (sec. Bory) and be related to the author’s taxonym with an exclude relationship. Another example is an “emend.” citation, which provides a more detailed circumscription of the taxon. Those should be treated as congruent potential taxa (if not stated otherwise by the author).

A source that cites a particular taxon name assigns a certain status to it, i.e. it uses it either as the name accepted for a taxon or as a synonym. Factual information can only be linked to accepted names (called correct names in the ICBN). The attribute StatusFk of table PTAXON points to the STATUS table and thus assigns the status of the used name (i.e., the status of the taxonym). STATUS is a catalogue table listing available values for the status. Apart from its primary key (StatusId) it contains the attributes Status (values see Table 16) and StatusAbbrev.

Table 16: The catalogue of values for the STATUS table

Status	Description
accepted (A)	the name is used as the correct (accepted) name of a taxon
synonym (S)	the name is mentioned as a synonym
partial synonym (P)	the name is mentioned as a partial synonym
pro parte synonym (L)	the name is mentioned as a pro parte synonym
unresolved (U)	there is still no decision on whether the name is used as an accepted name or as any kind of synonym

‘Partial synonym’ and ‘pro parte synonym’ are actually cases of concept synonymy stated in “traditional” treatments such as taxonomic monographs. When two or more taxa (with their respective types) are merged into a new taxon (with one of the old types as its type), the taxonyms corresponding to the old taxa are not merely synonyms to the taxonym corresponding to the new taxon but ‘partial synonyms’. This term and status was introduced for the Euro+Med project (GÜNTSCH & al., 2002). It describes a directed relationship to the accepted taxon from the synonym, which the system treats exactly as a “normal” synonymy. In reverse, every time a taxon is split in two or more new taxa, the old potential taxon may be cited as a pro parte synonym to all the new taxa (excluding the type except for one of the new ones). For this relationship, the status ‘pro parte synonym’ is assigned to the taxonym corresponding to the old taxon. As stated, concept relationships may be calculated from them and might replace them. However, defining them separately has the advantage that they can be distinguished from other concept relationships (e.g. to reconstruct the list of synonyms).

If the status assignation itself is not clear, this can be indicated using the DoubtfulFlag attribute, which (for accepted names) can also be interpreted as “provisional”. A further function of this attribute is to inactivate potential taxa (see discussion under “Changing the status from accepted to synonym” in GÜNTSCH & al., 2003).

Relationships between potential taxa

RELPTAXON (Table 17) contains all binary relationships that have been established between potential taxa or taxonyms represented by records in the PTAXON table.

Table 17: Attributes of the RELPTAXON table

Attribute	Type	Description
RelPTaxonId	int	Primary key for table RELPTAXON
PTNameFk1	int	Pointer to 1st potential taxon (name part)
PTRefFk1	int	Pointer to 1st potential taxon (reference part)
PTNameFk2	int	Pointer to 2nd potential taxon (name part)
PTRefFk2	int	Pointer to 2nd potential taxon (reference part)
RelQualifierFk	int	Pointer to RELPTQUALIFIER indicating the type of relationship
RelRefFk	int	Pointer to REFERENCE table providing the source of the assignation of this relationship

We distinguish three different areas of directed binary relationships between records of the PTAXON table: traditional synonymy, hierarchical taxonomic classification and concept synonymy. All three are stored in the RELPTAXON table because the formal structure of the relationship is the same: a first potential taxon (identified through the attributes PTNameFk1 and PTRefFk1) is related to a second one (identified through the attributes PTNameFk2 and PTRefFk2), and the relationship was established by a source (identified through the attribute RelRefFk). The relation itself is identified through the attribute RelQualifierFk, which points to the RELPTQUALIFIER catalogue table. That table holds all possible kinds of oriented relationships between two potential taxa in its attribute RelPTQualifier, its only attribute besides the primary key (RelPTQualifierId).

Basionym and homonym relationships are handled in the nomenclature part (see section on names above) and not in the potential taxon part of the model, because in principle they are not a matter of taxonomic opinion.

The area of "traditional" synonymy encompasses relationships commonly found in the lists of synonyms in taxonomic monographs or floras. Possible values of RelPTQualifier are: 'is synonym of', 'is partial synonym of', 'is pro parte synonym of' 'is misapplied name for', and, as a further specification of synonymy, 'is heterotypic synonym of' and 'is homotypic synonym of'. Note that the first three relations can only be established by a source (RelRefFk) that is the same as the circumscription reference of both taxonyms. In contrast, for the relationship 'is misapplied name for' the circumscription references must be different.

For classification relations we use the value 'is taxonomically included in'. Since classification is actually about taxa and not about names, classification relations are only meaningful for 'accepted' taxonyms. Different classifications can be distinguished through the authors of the classification relation (indicated by the attribute RelRefFk in the RELPTAXON table).

For concept relationships between potential taxa GEOFFROY & GÜNTSCH (2003) describe the 64 possible "combined relationships" (including the "doubtful" flag). As stated there, any "combined relationship" between two potential taxa PT_1 and PT_2 automatically has a corresponding reverse "combined relationship" between PT_2 and PT_1. Some concept relationships can also be automatically derived from other relations.

E. g. from classification relationships: in accordance with the rules of nomenclature, the taxa (with their accepted or correct name and their rank) in a single treatment (source) form a tree in which the nodes (potential taxa) of different branches do not overlap. For example, a subspecies of a certain species cannot have elements in common with any infraspecific taxon of another species, nor with others of the same species itself. This also implies that taxa of the same rank cannot overlap within the same treatment. These are thus implied 'excludes' relationships. The relationship 'is taxonomically included in' implies an 'is included in' concept relationship and its reverse. Synonym relationships (hetero- or homotypic) generally imply an 'overlap' (but only rarely 'congruent') relationship with the potential taxon circumscribed by the original publication of the synonym; for a given group of homotypic synonyms, this can be established between all members of the group.

Data integrity rules for the concept-section of the model

Rank-dependent integrity rules

- aggregates and species groups may not exist on their own, there must be at least two species ("microspecies") linked to an aggregate by means of an 'is taxonomically included in' relationship
- one of these "microspecies" must carry the same combination of genus name and species epithet as the aggregate
- traditional synonym relations are normally only established between the following rank categories: species and below; above species to genus; and above generic level.

Most status-dependent integrity rules are given in Table 18.

- if a taxonym has the status 'pro parte synonym' then there must be at least two traditional synonymy relations (of the kind 'is pro parte synonym of') of it with at least two different 'accepted' taxonyms.
- if a taxonym has the status 'partial synonym' then there must be at least two relations of the kind 'is pro parte synonym of' with the same 'accepted' taxonym.
- if the relation between two potential taxa is 'is taxonomically included in' then both of them must have status 'accepted', the rank of the first taxonym must be lower than that of the second one, and no other classification relation may be made by the same source for the same first taxonym (i.e. the combination of the attributes PTNameFK1, PTRefFK1 and RelRefFk in the RELPTAXON table is unique for the relationship 'is taxonomically included in'. NB: an exception could be made here for relationships with aggregates.

Table 18: Status dependent records and values of attributes in table RelPTaxon
(● = mandatory, i.e. the first taxonym must have such a relationship defined in RelPTaxon, ○ = optional, ✖ = prohibited, × = not applicable, = = same, ≠ = different)

First taxonym (PTNameFk1, PTRefFk1) has	and is	Second taxonym (PTNameFk2, PTRefFk2)				
		Status 'A' active	Status 'A' inactive	Status 'S', 'P', or 'L'	PtRefs	Source Ref.
Status 'A' active	synonym of	✖	✖	✖	×	×
	partial synonym of	✖	✖	✖	×	×
	pro-parte syn. of	✖	✖	✖	×	×
	misapplied name for	○	○	✖	≠	=Pt2
	taxon. included in	●[1]	✖	✖	=/≠	=Pt1
	concept synonym of	○	○	✖	≠	=/≠
Status 'A' inactive	synonym of	✖	✖	✖	×	×
	partial synonym of	✖	✖	✖	×	×
	pro-parte syn. of	✖	✖	✖	×	×
	misapplied name for	✖	✖	✖	×	×
	taxon. included in	○[2]	○	✖	=/≠	=Pt1
	concept synonym of	✖	○	✖	≠	=/≠
Status 'S', 'P', or 'L' active	synonym of ('S' only)	●	✖	✖	=	=
	partial synonym of ('P')	●	✖	✖	=	=
	pro-parte syn. of ('L')	●	✖	✖	=	=
	misapplied name for	✖	✖	✖	×	×
	taxon. included in	✖	✖	✖	×	×
	concept synonym of	✖	✖	✖	×	×
Status 'S', 'P', or 'L' inactive	synonym of ('S' only)	✖	●	✖	=	=
	partial synonym of ('P')	✖	●	✖	=	=
	pro-parte syn. of ('L')	✖	●	✖	=	=
	misapplied name for	✖	✖	✖	×	×
	taxon. included in	✖	✖	✖	×	×
	concept synonym of	✖	✖	✖	×	×
Status 'U' active or inactive	synonym of	✖	✖	✖	×	×
	partial synonym of	✖	✖	✖	×	×
	pro-parte syn. of	✖	✖	✖	×	×
	misapplied name for	✖	✖	✖	×	×
	taxon. included in	○	○	✖	=/≠	=Pt1
	concept synonym of	✖	✖	✖	×	×

[1] Except at the highest hierarchical level

[2] This is an exception to the rule that inactive parts of the tree should not be related to active ones; the relationship 'inactivated' is used to retain the inclusion in a taxonomic group for access and sorting purposes

4. Factual information: the facts section

Factual data in biology is inherently complex and often described in separate information domains, e. g. ecological data (e.g. PEET & al., 1998), descriptive information (e.g. HAGEDORN, 1999, and DIEDERICH & al., 1997), analytical data (see BERENDSOHN & al., 1997, for an example from caryology), or geographic distribution information. All these should always be linked to potential taxa, but these domains clearly form either separate systems or extensions to the Berlin Model.

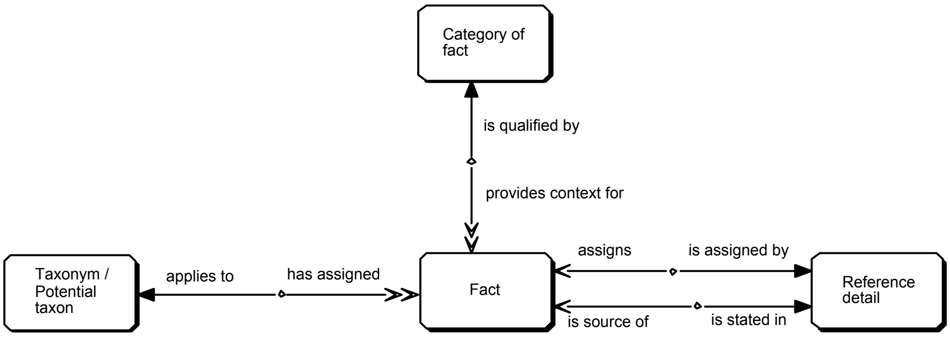


Figure 6: Factual data

To demonstrate the linking of factual information, the core model supports simple facts consisting of only one value and a fact category (Figure 6, Table 19).

Table 19: Attributes of table FACT

Short name	Type	Description
FactId	int	Primary key for table FACT
PTNameFk	int	Pointer to PTAXON table (name part)
PTRefFk	int	Pointer to PTAXON table (reference part)
Fact	text	One fact's value as free text (e.g. ‘Daisy’ for the fact category ‘Common name’)
FactCategoryFk	int	Pointer to FACTCATEGORY table
FactRefFk	int	Pointer to REFDETAIL (combined foreign key), indicating the source of the fact
FactRefDetailFk	int	
PTDesignation-RefFk	int	Pointer to REFDETAIL (combined foreign key), indicating the reference for the assignation of the fact to the potential taxon
PTDesignation-RefDetailFk	int	
DoubtfulFlag	bool	Set for doubtful assignation of fact to the potential taxon

Any other data structure in a separate table or system of tables can be linked in the same way to a potential taxon. An example is the treatment of geographical distribution and occurrence data in the Euro+Med PlantBase project (see Berendsohn & al., 2002, for details). The table OCCURRENCE provides the presence status of a taxon in a certain area (e.g. 'native' or 'introduced'). It is linked to the PTAXON table in the core, and to a table AREA holding the standard areas defined for the project. It is also linked to the reference section, so that every record's source can be documented.

Alternatively, it is possible to use a specific fact category to link such "external" systems to potential taxa, using the FACT table as intermediate (the table FACTCATEGORY contains just the primary key and the attribute FactCategory).

Factual data can only be linked to accepted taxa. Handling the transfer of factual data from existing sources to new treatments by experts is one of the main problems to be solved by the taxonomic editor software (GÜNTSCH & al., 2003). Developing tools to handle the automatic linking of factual information was the foremost objective of the MoReTax project (see GEOFFROY, 2003), which devised a "transmission engine" to be put into place in forthcoming projects.

Conclusion

Using a concept-based system has two main purposes. On the one hand, it provides taxonomists with a tool to store their decision processes in a formal way and thus helps to make taxonomic information falsifiable. On the other hand, for a much broader user community it will make the linking of biological data by means of scientific names a more reliable tool to integrate biological information. We hope that Euro+Med, AlgaTerra, EuroCat, GBIF and other projects will help further develop this concept as part of a broad international collaboration.

References

BERENDSOHN, W. G. (1994): IOPI vascular plant checklist. A CASE model of checklist system data, version 6.0. In: WILSON, K. (ed.), Global Plant Checklist project plan. International Organisation for Plant Information, Sydney. Version 7.3. [23 Jun 1995]: http://www.bgbm.fu-berlin.de/iopi/iopimodel73/7301root.htm.

BERENDSOHN, W. G. (1997): A taxonomic information model for botanical databases: the IOPI model. Taxon 46: 283-309.

BERENDSOHN, W. G. (in press): ENHSIN in the context of the evolving global biological collections information system. The Natural History Museum, London.

BERENDSOHN, W. G., GREILHUBER, J., ANAGNOSTOPOULOS, A., BEDINI, G., JAKUPOVIC, J., NIMIS, P. L. & VALDÉS, B. (1997): A comprehensive datamodel for karyological databases. Plant Syst. Evol. 205: 85-98.

BERENDSOHN, W. G., GEOFFROY, M., GÜNTSCH, A. & LI, J. (2002 [3 Feb 2003]): The Berlin Taxonomic Information Model. - http://www.bgbm.org/biodivinf/docs/bgbm-model/.

BRUMMITT, R. K. & POWELL, C. L. (1992): Authors of Plant names. Royal Botanic Gardens, Kew.

DIEDERICH, J., FORTUNER, R. & MILTON, J. (1997): Construction and integration of large character sets for Nematode morpho-anatomical data. Fundam. appl. Nematol. 20: 409-424.

ELANKOVAN, S., BERENDSOHN, W. G. & MEYER, H. (1996 [10 Feb 2003]): Person Teams in the CDEFD and IOPI models: an implementation. http://www.bgbm.org/CDEFD/PersonTeams/Title.htm.

GEOFFROY, M. (2003): Towards the implementation of the "Transmission Engine". Schriftenreihe Vegetationsk. 39: 87-112.

GEOFFROY, M. & BERENDSOHN, W. G. (2003a): The concept problem in taxonomy: importance, components, approaches. Schriftenreihe Vegetationsk. 39: 5-14.

GEOFFROY, M. & GÜNTSCH, A. (2003): Assembling and navigating the potential taxon graph. Schriftenreihe Vegetationsk. 39: 71-82.

GREUTER, W., MCNEILL, J., BARRIE, F. R., BURDET, H. M., DEMOULIN, V., FILGUEIRAS, T. S., NICOLSON, D. H., SILVA, P. C., SKOG, J. E., TREHANE, P., TURLAND, N. & HAWKSWORTH, D. L. (2000): International Code of Botanical Nomenclature (Saint Louis Code) adopted by the Sixteenth International Botanical Congress St. Louis, Missouri, July - August 1999. Regnum Veg. 138: 1-474. [Also available in electronic format under http://www.bgbm.org/iapt/nomenclature/code/.]

GÜNTSCH, A. LI, J. & BERENDSOHN, W.G. (2002 [Jun]): Design of the Internet Taxonomic Sector Editor; Version 1.5. http://www.bgbm.org/BioDivInf/Projects/Euro+Med/EditorDesign.pdf.

GÜNTSCH, A., GEOFFROY, M., DÖRING, M., GLÜCK, K., LI, J.-J., RÖPERT, D., SPECHT, F. & BERENDSOHN, W. G. (2003): The taxonomic editor. Schriftenreihe Vegetationsk. 39: 43-56.

HAGEDORN, G. (1999 [10 Feb 2003]): DeltaAccess - a SQL interface to DELTA (Description Language for Taxonomy), implemented in Microsoft Access [User guide and documentation]. http://160.45.63.11/Workbench/Descriptions/Docu160/DELTAACCESS_IN.html.

KUSBER, W.-H., GLÜCK, K., GEOFFROY, M. & JAHN, R. (2003): Typification – an extension of the Berlin Taxonomic Information Model. Schriftenreihe Vegetationsk. 39: 57-70.

LAWRENCE, G. H. M., BUCHHEIM, A. F. G., DANIELS, G. S. & DOLEZAL, H. (1968) (eds.): B-P-H, Botanico-Periodicum-Huntianum. Hunt Botanical Library. Pittsburgh.

PANKHURST, R. J. (1993): Taxonomic Databases: The PANDORA System. In: FORTUNER R. (ed.): Advances in computer methods for systematic biology: Articificial Intelligence, databases, computer vision. John Hopkins University Press.

PEET, R. K., WENTWORTH, T. R. & WHITE, P. S. (1998): A flexible, multipurpose method for recording vegetation composition and structure. Castanea 63: 262-274.

STAFLEU, F. A. & COWAN, R. S. (1976-1988): Taxonomic Literature, 2nd. ed., vol. 1-7. Regnum Veg. 94, 98, 105, 110, 112, 115 & 116.

STEVENS, W. D., ULLOA ULLOA, C., POOL, A. & MONTIEL, O. M. (2001): Flora de Nicaragua. Monogr. Syst. Bot. Missouri Bot. Gard. 85. Missouri Botanical Garden Press.

TREHANE, P., BRICKELL, C. D., BAUM, B. R., HETTERSCHEID, W. L. A., LESLIE, A. C., MCNEILL, J., SPONGBERG, S. A. & VRUGTMAN, F. (1995): International Code of Nomenclature for Cultivated Plants. [ICNCP or Cultivated Plant Code.] Quarterjack Publishing, Wimborne.

The taxonomic editor

ANTON GÜNTSCH, MARKUS DÖRING, MARC GEOFFROY, KARL GLÜCK, JINLING LI, DOMINIK RÖPERT, FRANK SPECHT & WALTER G. BERENDSOHN

Four basic activities are involved in the taxonomic editing process required for the MoReTax project. First, the editor software has to support data entry and editing of data referring to scientific names, their authors, their corresponding nomenclatural reference citations, their nomenclatural status (e.g. validity, legitimacy), and their relationships to other names (e.g. basionyms, homonyms, or orthographic variants). Second, the editor has to support capture, correction, and readjustment of taxonomic opinion as expressed by a specific reference (a person or a publication – monograph or Flora). This implies relating taxa to other taxa (e.g. synonym relationships), and building taxonomic trees by linking taxonomic subgroups to their respective parents. Third, the editor has to support the display, adding, and editing of factual information associated with taxa, such as geographic distributions or threat status values. Fourth, the editor should enable taxonomists to establish relationships between arbitrary taxonomic concepts thus building the ground for propagating factual data through networks of potential taxa as described in GEOFFROY & GÜNTSCH (2001, 2003) and building interfaces having the ability to rank the reliability of information presented to users seeking taxonomic knowledge in biological information systems.

There are a variety of taxonomic editors available providing specific solutions for the first three tasks, on diverse technological platforms. To name three: the Bibmaster system (PANDO, 2000) implemented with Microsoft Access is a taxonomic editor that allows for convenient rapid input of taxon lists and references and cleaning up the data entered. Inconsistency detection mechanisms are implemented based on standard author and reference lists. The Advanced Revelation (REVELATION, 2002) based Pandora system (PANKHURST, 1993) is implemented on Microsoft DOS and provides a great degree of data integrity being built on a strictly hierarchical data structure for scientific names. Finally the Euro+Med Remote Editor (GÜNTSCH & al., 2002) is a World Wide Web tool allowing taxonomic experts to carry out taxonomic revisions online on a centralised database without having to implement additional software or a local database system on their computer. The editor is under development and will be made available in 2003.

The Euro+Med editor is built on the concept-based "Berlin Taxonomic Information Model" (BERENDSOHN & al., 2003) and it allows the input of basic concept relations to known concepts (e.g. declaring the treatment of a taxon to be identical to that in the original Flora Europaea, or to a newer one in a more recent revision or Flora). The "Berlin Model" will also form the basis for the MoReTax system (and the following references to tables and attributes are drawn from that source), and the Euro+Med editor will be used to depict some forms. Nevertheless, the Euro+Med editor is a project-specific implementation and does not yet represent a full-blown taxonomic editor covering concept editing. The text of this article represents our current state of knowledge with respect to the specification of a software tool fulfilling the broad scope of requirements set for an editor within the taxonomic information system envisioned by MoReTax.

Technical considerations

The MoReTax taxonomic editor will be based on a relational database system implementing the Berlin Model. Object-oriented models and systems may have their advantages, but the pragmatic view of BERENDSOHN (1997) still prevails: using a mainstream solution at the database level is imperative for a relatively small and under-financed user community such as ours. The system will also be based on a centralised database management server rather than relying on the distribution of an application based on a small-scale database system such as Microsoft Access. This approach will minimise the need for reintegration of shared data sets such as references and scientific names. The latest version of such authority files will always be immediately visible to all users of the system. Moreover, the software will be implemented on an application server (see Figure 1), e.g. using Java server pages (SUN, 2003) or ColdFusion (MACROMEDIA, 2003), i.e. the client software will not be programmed as an application that has to be installed by users on their own computers. Instead, standard World Wide Web browser software will suffice to carry out taxonomic work on any operating system platform.

The browser must provide certain basic requirements such as Java-Script execution needed to perform rudimentary form field validations.

With this approach, an expert network can easily be set up without having to recompile software for various system setups without need for redistribution in case of updates. However, due to the stateless nature of the http protocol, this decision has significant consequences for style and implementation of the application.

A stateless communication protocol does not maintain a connection between the client and the server. A data input form displayed thus does not "know" in which context it has been called from the server. Using http, states have to be simulated with the help of parameterised URLs or server variables corresponding to cookies on the client computer. This presents an obstacle to quick "drag and drop-like" prototyping of user interface forms. On the other hand, html based user interfaces are well suited for programming dynamic forms changing their appearance depending on the content to be displayed.

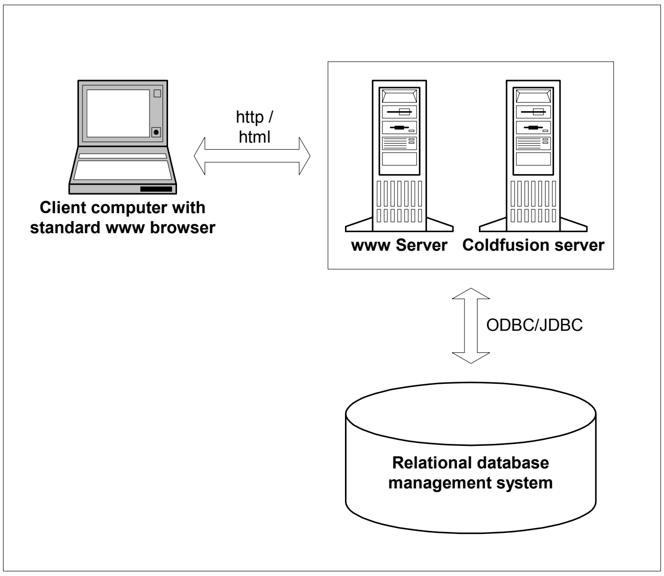

Client computer with standard www browser
http / html
www Server
Coldfusion server
ODBC/JDBC
Relational database management system

Figure 1: Taxonomic editor example architecture using ColdFusion

Basic integrity rules (e.g. “there is no name without a rank”) will be implemented at database level using the DBMS’s ability to define constraints such as referential integrity rules, data typing and default values. BERENDSOHN & al. (2003) discussed further constraints (“data integrity rules”) in the context of the information model. Some of these can be circumvented by avoiding certain inputs that could generate an error (for example not allowing to change the rank of a name). Others will have to be enforced immediately by dialogs with the user in the case of errors, but this has to be kept at a minimum to avoid user frustration – e.g. for cases where a preliminary input is necessary. An alternative is to provide routines that perform checks and inform the user (or a subsequent editor) of such errors.

Database access (insertions, selections etc.) should be encapsulated within stored procedures to hide internal database structures from the client application as much as possible. Regularly recurring operations consisting of several individual SQL statements should also be wrapped into stored procedures so that they appear as a single operation.

For example, the correction of names at user interface level initiates a sequence of operations at database level: the name record to be updated is moved to the NAMEHISTORY table, foreign keys referencing author teams or nomenclatural citations are replaced with the text content they are pointing to, and the SuccNameHistory attribute is set to keep track of the editing history; finally, the corrected scientific name is inserted in the NAME table. These operations should be processed in a single reusable procedure implemented at database level.

All messages and prompts should not be hard-coded within programs or stored procedures but should be stored in database tables. This will greatly facilitate later internationalisation, because warnings, form headers, field labels, help texts, and the like can easily be translated without changing the program code.

Certain complex operations such as parsing of free text name or reference fields should not be implemented at database level because they are best programmed using powerful object oriented languages such as Java rather than the cumbersome languages provided by database management systems. These functions may be bundled and provided as a class library for various applications and user interfaces.

Basic user interface design and navigation

The complex nature of taxonomic information poses a challenge for the design of a transparent and user-friendly interface.

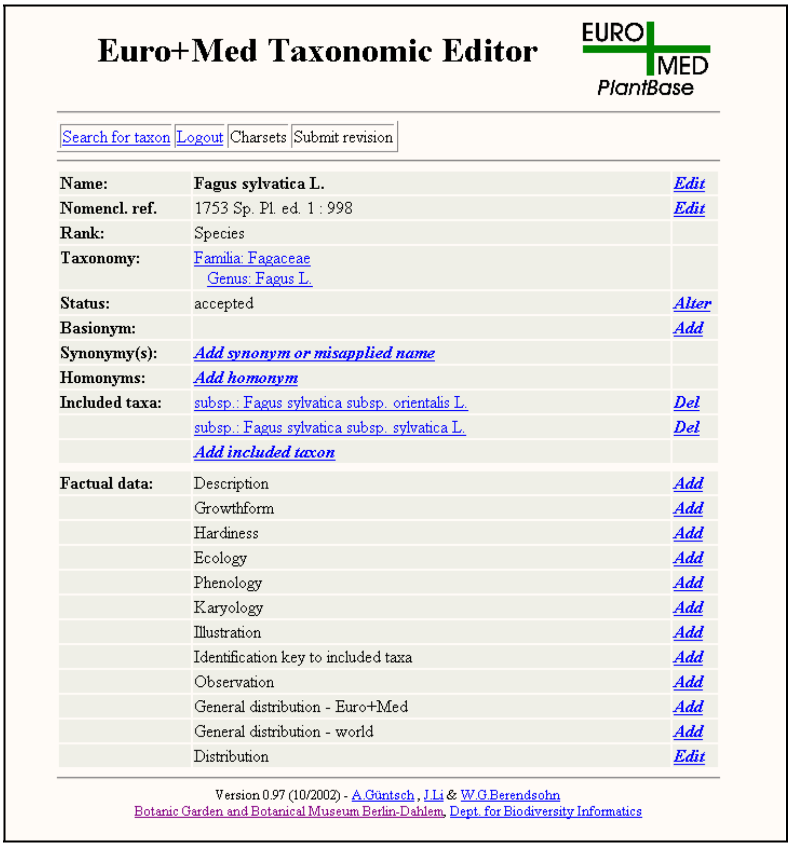
Taxonomic information consists of multiply interconnected information elements. Scientific names, references, potential taxa, synonyms, etc. form a highly complex network of relationships, as illustrated by the Berlin Model and the corresponding potential taxon graph (GEOFFROY & GÜNTSCH, 2003).

The user should be given access to navigate and edit the full information content of the system. At the same time, forms should be kept simple and should not present too much information – but inconvenient navigation through long sequences of interlinked forms should also be prevented.

These seemingly conflicting demands can be met with an approach based on a single central form, which always focuses on a single potential taxon record (accepted taxon or synonym).

Summaries of information are shown, such as the nomenclatural reference citation in concatenated form, lists of synonyms and related names (e.g. the basionym), higher taxa and those enclosed, and links to factual information pertaining to the displayed taxon. The form adapts to its content. So, for example, a synonym list and the function to change that list are only shown for accepted taxa. The form allows navigating through the potential taxa in the system by means of the hyperlinks displaying higher taxa, enclosed taxa, synonyms (or the accepted name), etc. The Euro+Med Internet taxonomic editor exemplifies this approach with its central navigation form (see Figure 2).

The central form also provides functions to edit the entries themselves (e.g. the nomenclatural reference citation for a name, or the status of the taxon), or to edit the relationships the potential taxon has with others (e.g. by adding or removing a synonym). Other forms are called to perform these changes, but deep nesting of forms is avoided – normally, the user needs to access only the form directly called from the central form to execute the operation and return to its starting point.



Euro+Med Taxonomic Editor

Name:	**Fagus sylvatica L.**	Edit
Nomencl. ref.	1753 Sp. Pl. ed. 1 : 998	Edit
Rank:	Species	
Taxonomy:	Familia: Fagaceae Genus: Fagus L.	
Status:	accepted	Alter
Basionym:		Add
Synonymy(s):	Add synonym or misapplied name	
Homonyms:	Add homonym	
Included taxa:	subsp.: Fagus sylvatica subsp. orientalis L.	Del
	subsp.: Fagus sylvatica subsp. sylvatica L.	Del
	Add included taxon	
Factual data:	Description	Add
	Growthform	Add
	Hardiness	Add
	Ecology	Add
	Phenology	Add
	Karyology	Add
	Illustration	Add
	Identification key to included taxa	Add
	Observation	Add
	General distribution - Euro+Med	Add
	General distribution - world	Add
	Distribution	Edit

Version 0.97 (10/2002) - A.Güntsch , J.Li & W.G.Berendsohn
Botanic Garden and Botanical Museum Berlin-Dahlem, Dept. for Biodiversity Informatics

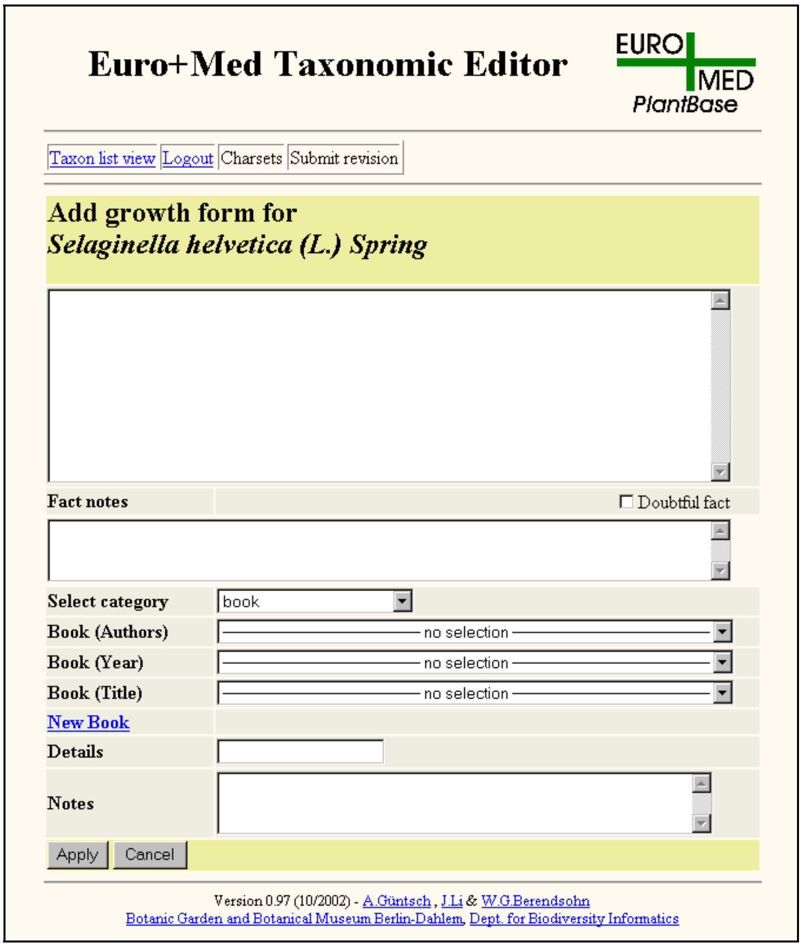
Figure 2: Central navigation form of the Euro+Med editor

Editor functions that are independent of the focused taxon (e.g. search on names) are placed within the header area of the form and remain unchanged irrespective of their context.

Apart from the above mentioned navigation features of the form itself, access to the form can be gained either by searching for a name, or by means of two checklist views, representing either a synonymised list of potential taxa or an alphabetical list of taxonyms. Every entry in the list again forms a hyperlink to the respective central form. The checklist views can be further restricted by filter conditions: restrictions on parts of the name, on a particular higher taxon, or on a particular circumscription reference are possible.

Editing facts

The core Berlin Model provides just a simple table FACT basically consisting of a free text field, a key indicating the factual data type stored in FACTCATEGORY, and the source reference for that fact. It thus accommodates factual data of any kind as text without enforcing further structure. This was implemented in Euro+Med for several of the required data items, while others (e.g. the standardised geographic distribution and distribution status) are accommodated by project specific extensions.



Euro+Med Taxonomic Editor

Add growth form for
Selaginella helvetica (L.) Spring

Fact notes □ Doubtful fact

Select category	book
Book (Authors)	no selection
Book (Year)	no selection
Book (Title)	no selection
New Book	
Details	
Notes	

Version 0.97 (10/2002) - A.Güntsch , J.Li & W.G.Berendsohn
Botanic Garden and Botanical Museum Berlin-Dahlem, Dept. for Biodiversity Informatics

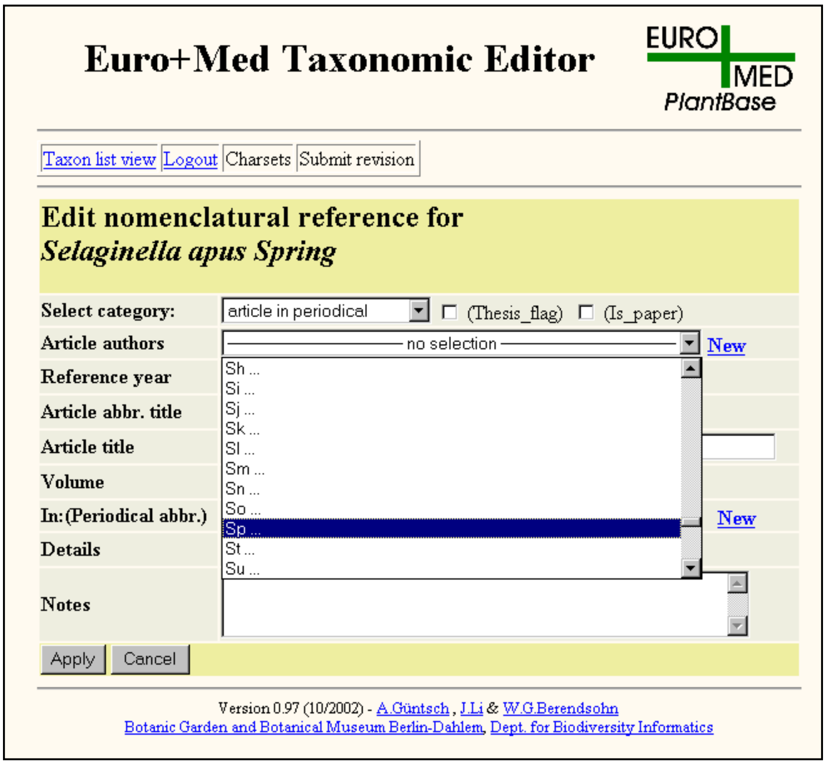
Figure 3: Adding a fact and its bibliographic reference

Since facts may only be linked to taxonyms in the model (in fact, only to accepted potential taxa), the set of factual data editing forms can be linked to the central navigation form. An *add* or an *edit* button is displayed depending on the existence of previously entered or imported facts (Figure 2). The form called by the link is identical for each factual data type and provides access to the data items mentioned (Figure 3).

As stated, for some types of factual data the representation as a free text field is inadequate, especially if the information is used for further processing (e.g., geographic data used to create distribution maps). In this case forms have to be designed individually on the basis of the data structure chosen to capture the information.

References

The Berlin Taxonomic Information Model treats references of any kind (e.g. books, articles in journals, published CD ROMs, websites) with a single recurrent data structure capable of properly representing the often nested structure of references (e.g. "article in periodical" or "part of a book"). Although designing a single form for all reference types would probably be possible, it is recommended to assign the task of entering references to two forms, one for nomenclatural references and the other for bibliographic references. This is justified because the set of fields vary partly, and full reference strings concatenated to monitor data entry differ significantly.



Euro+Med Taxonomic Editor

Edit nomenclatural reference for
Selaginella apus Spring

Select category: article in periodical □ (Thesis_flag) □ (Is_paper)
Article authors: no selection
Reference year
Article abbr. title
Article title
Volume
In:(Periodical abbr.)
Details
Notes

Version 0.97 (10/2002) - A.Güntsch , J.Li & W.G.Berendsohn
Botanic Garden and Botanical Museum Berlin-Dahlem, Dept. for Biodiversity Informatics

Fig. 4: Editing a nomenclatural reference

The reference forms should implement all reference categories specified with the REFCATEGORY table in a single template only displaying the relevant fields for the selected reference category. For example both the 'database' and the 'published CD ROM' reference types have a title whereas export date is applicable to database references only. If a reference is part of another title the parent title should be picked from a select box and not retyped by the user. As with all select box fields this may cause performance problems particular for book titles, which may consume hundreds of characters each. Therefore, it is recommendable to implement a set of coupled fields (authors, year of publication, title) so that, for example, pre-selecting an author team reduces the number of book titles offered.

The nomenclatural reference form will use standardised abbreviated titles for selecting full book or journal titles. A field for selection of standardised nomenclatural author teams allows entering the authors for "in"-citations (Figure 4).

A reference form may appear by itself (e.g. linked to the central form for entering the nomenclatural reference of a name) or as part of other forms (e.g. for adding a bibliographic record to a fact). Similar to names, a function for identifying duplicate references should be implemented and warn users trying to enter already existing ones.

In many cases users will not find journal or book titles needed to construct a reference within in the pick lists generated from the databases REFERENCE table because the title has not yet been imported or it is belonging to a recent publication. Therefore, an additional form should be linked which can be used to enter new titles.

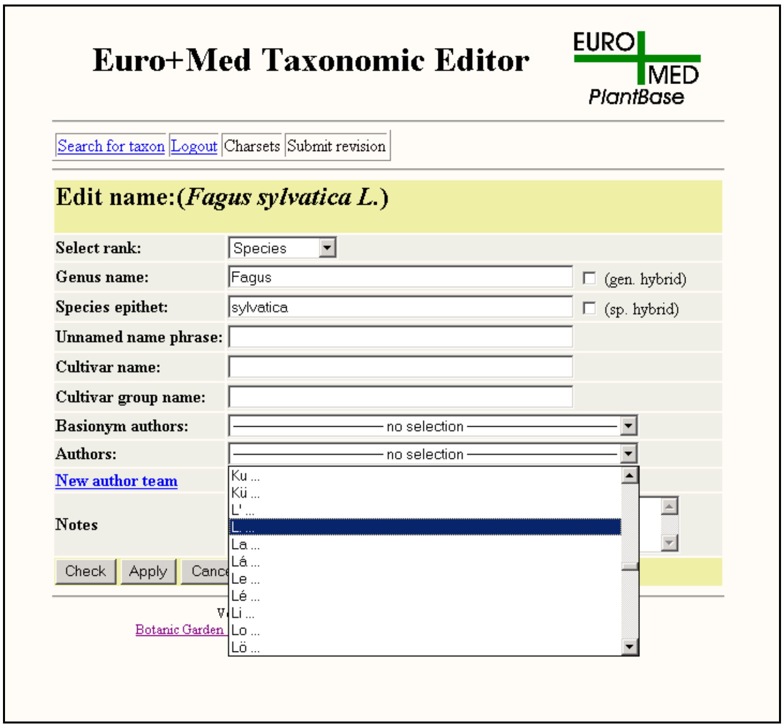
Editing names

The layout of the form used for adding and editing names partly depends on the rank of the name. In accordance with the selection the fields appropriate for the rank (e.g. genus name and specific epithet for a species) are displayed and can be filled or changed. A syntax check should be provided giving users the means to check whether basic nomenclatural rules are fulfilled (e.g. "is the genus name a string consisting of an upper case character followed by a sequence of lower case characters, free of blanks?"). An additional select box allows indicating the nomenclatural status of a name (e.g. "nomen conservandum").

With existing names, the rank should normally not be changed, because this would require extensive checking of rank-dependent information in related entities. For example, relationships of the potential taxon to higher taxa and included taxa would have to be checked for the consistency of the taxonomic tree.

The Berlin Model provides cache fields within several tables used to capture pre-calculated results to avoid time-consuming re-computation of information repeatedly used at user interface level (e.g. the FullNameCache in table NAME, containing the full Latin name as a concatenation of atomised name parts). This concatenation is normally executed by a trigger upon detection of changes of the underlying atomised data elements. But these cache fields may also be used to house preliminary or incomplete data entered by users or imported from data sources providing a low degree of atomisation. They are than protected from being overwritten by the setting of the PreliminaryFlag. At the interface level, an additional free text field should be added to the form to display preliminary data and to give the user the choice of entering both structured and unstructured data. Removing the flag by the corresponding button then releases structured data once they are considered complete and correct.

Author teams (basionym and combination authors) should be made available as select boxes rather than entered as free text. This is to ensure compliance with author teams and the abbreviated author names stored in the database. However, offering extensive lists of catalogue data in select boxes is one of the major performance bottlenecks in wide area web based editor systems. Therefore, a two-step selection mechanism is implemented allowing to pre-select from a list of character combinations and then picking from the list of matching author teams (Figure 5).



Euro+Med Taxonomic Editor

Edit name:(*Fagus sylvatica L.*)

Select rank: Species

Genus name: Fagus ☐ (gen. hybrid)

Species epithet: sylvatica ☐ (sp. hybrid)

Unnamed name phrase:

Cultivar name:

Cultivar group name:

Basionym authors: no selection

Authors: no selection

Notes

Figure 5: Reducing "data traffic" with pre-selections

If an author team that is needed for a name does not exist in the underlying AUTHORTEAM table, the entry has to be created by the user with the help of a separate form providing a series of select boxes each containing the list of standard author name abbreviations used to create an ordered sequence of authors. Author names should also be selected with a two-step mechanism to decrease the amount of data needed to populate the form. Again, the author team cache field can be used for preliminary entries if users do not want or are unable to create structured records.

Finally, the name form provides functions to enter hybrids. Named hybrids (nothotaxa) can be entered with the fields already provided for non-hybrid names. The hybrid markers are not included in the genus name or the epithet but will be added automatically by a trigger function when concatenating the full name. A hybrid tick box for the relevant name elements serves to indicate that the name is a named hybrid. Unnamed hybrids (hybrid formulas) are indicated by checking a tick box and subsequently constructed by selecting the parental taxon names from the name table.

When creating a new name, there is always a certain risk of duplication. To minimise the probability creating duplicates, a function should be implemented that finds potential duplicates by means of an appropriate similarity measurement for name strings. Since this function will be beneficial for several different applications, it should be implemented at database level.

The Berlin Model provides for relationships between names (as opposed to relationships between taxa, see below). Although these can be accessed directly from the central form, they should also be accessible from the name form. Consequently, lists of later homonyms and (in the case of combinations) a basionym or substituted synonym are depicted and can be edited, added, or removed from the name entry form.

Creating potential taxa and taxonyms

Entering a name and creating a taxonym are two different operations, but this is a separation, which – at least from the point of view of a revising author – looks artificial. In the case of the Euro+Med editor, the approach is based on the premise that the user is the author of a treatment within the database. A copy containing a taxonomic "slice" of the general database (e.g., a particular family) is generated at the outset of the revision. For all the taxonyms in that copy, the revising author is assigned as the circumscription reference of all taxonyms, classification relations, and conventional synonymic relations between the potential taxa within the treatment. Authors can then freely edit and change these, thus creating a consistent treatment in accordance with their taxonomic opinion.

In its initial phase, the principal aim of the Euro+Med project is to provide a single synonymised taxonomic checklist of the European flora, so concept relations are presently secondary in importance. However, relations to the treatment in the printed Flora and/or a reference to a congruent taxonomic concept provided by other references may be created, but this is always done starting from the author's own treatment.

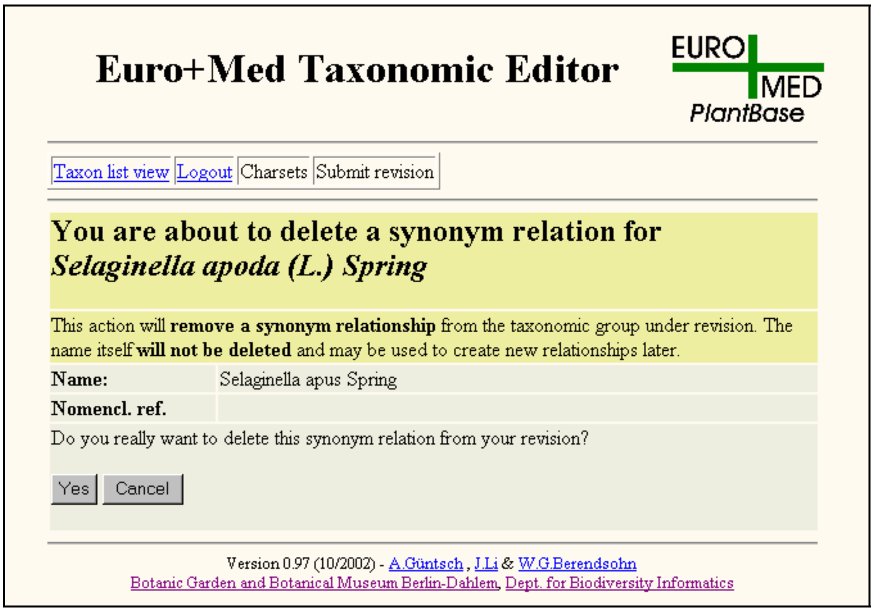
Although this procedure will prevail for users who are authors of taxonomic treatments, the editor here described will have to support managerial and rapid data entry functions. For example, it must be possible to rapidly enter a literature revision of a group based on names existing in the database. In a preliminary form, this can be done by a form providing an entry for the circumscription reference used (i.e., the revision) and listing relevant names, offering in a first step to assign accepted status to each of them. In a second step, synonym status can be assigned to remaining names, coupled with a selection of the accepted name from the taxa defined in the first step. Finally, accepted names can be arranged in a classification, i.e. included taxa can be assigned to them from the pool of accepted names of lower rank and appropriate name elements (to ensure that a species carries the appropriate generic name, etc.). For the future, we envisage using a tool where the existing names are represented by hyperlinked entries in a taxonomic tree, and where their status and classification can be arranged by drag and drop techniques. However, since this requires intense checking of data integrity rules and the display of - and navigation in – potentially large views of the database, proper techniques to implement this as a web-based application still have to be developed. This is particularly the case if concept relations are to be included. Further research is needed to develop a proper specification for these parts of the editor.

Editing taxon relationships

Taxonomists traditionally distinguish three kinds of taxon relationships: synonymy, classification, and (in the form of notes about misapplications or references to “sensu stricto” or “sensu lato” concepts) concept relations. The Berlin Model supports all three through the RelPTaxon table and the relationship types defined in the accessory RELPTQUALIFIER table. Data integrity rules referring to the partners possible in such a relationship are fixed by assigning a status to a taxonym and, in the case of classifications, by the rank of the partners.

Synonym relationships can be established using the central form as demonstrated by the Euro+Med editor. All names that have not been used as a taxonym with the author’s circumscription reference are available for assignation. When chosen, they are automatically becoming a taxonym record of the author’s with synonym status assigned and the relation entered.

This combination of several operations into a single user function deserves some consideration in the design. On the one hand, the relatively complex procedure of entering a name record, creating a potential taxon, and adding entries to the RELPTAXON table is hidden and simplified for the user. On the other hand, subsuming multiple operations in a single function bears the danger of “uninformed” users, who have no clue about the consequences of their actions. Therefore, all forms should inform about the kind of data and relations created, modified, or deleted. Figure 6 shows the Euro+Med form for deleting synonyms as an example. The header informs the user that this action will not delete the name but a previously existing synonym relation to an accepted (potential) taxon.



Euro+Med Taxonomic Editor

You are about to delete a synonym relation for *Selaginella apoda (L.) Spring*

This action will **remove a synonym relationship** from the taxonomic group under revision. The name itself **will not be deleted** and may be used to create new relationships later.

Name:	Selaginella apus Spring
Nomencl. ref.	

Do you really want to delete this synonym relation from your revision?

Version 0.97 (10/2002) - A.Güntsch, J.Li & W.G.Berendsohn
Botanic Garden and Botanical Museum Berlin-Dahlem, Dept. for Biodiversity Informatics

Figure 6: Deleting a synonym

For classification and concept relationships, only taxonyms of accepted status are available. For classification in another potential taxon, all accepted taxonyms of higher rank can be assigned (assignation of genera and species to lower taxa must follow the appropriate name integrity rules). As a rule, for the selection of taxa to be included only those of the author's circumscription reference should be offered for selection (again respecting the appropriate rules when the taxon to be included is of infrageneric rank).

Changing the status of a taxonym

Building a comprehensible and convenient user interface for altering status values of taxonyms is challenging because status changes may influence considerable parts of the potential taxon graph (e.g. a taxon and its entire sub tree of included taxa) and its connected set of factual data items.

The Berlin Model presently recognises 3 basic values for the status of the taxonym: 'accepted', 'synonym', and 'unresolved'. An additional attribute can be set to modify the status assignment, indicating provisionally accepted names, doubtful synonyms, or those records that are in a transitional state (e.g. those left over after their parent taxon has been deleted or changed status to synonym, see below). The latter one is not directly set. A simple select box within the respective editing form can set the provisional/ doubtful state because it is used only in output.

The specifications for the Euro+Med editor called for two more status values: 'partial synonym' and 'pro-parte synonym'. These are in fact concept relationships and will not be discussed further in this context.

Changes between synonym or unresolved status on the one hand and accepted status on the other may be rather complex and must therefore be considered in detail.

Changing the status from synonym to accepted

Changing the status attribute of a taxonym from 'synonym' to 'accepted' implies that the synonym relation between this taxonym and its linked accepted potential taxon must be severed. Since the Berlin Model does not allow you to link factual data, included taxa, and further synonyms to a synonym, no additional modifications of the potential taxon graph have to be applied. However, one consequence of this operation is that the resulting accepted taxon is isolated in a sense that it is not linked to a higher taxon. Consequently, it is not possible to focus on this taxon anymore by navigating through the tree of taxa to be edited. An intermediate form should prompt the user to select a higher taxon before altering the status. Alternatively, the potential taxon status may be set to "inactive" to allow for later processing.

Changing the status from accepted to synonym

Factual data, synonyms and included potential taxa can only be linked to accepted potential taxa. Therefore, a status change from 'accepted' to 'synonym' involves significant modifications of the affected parts of the potential taxon graph (Figure 7). Initially, the user has to decide to which accepted taxon (Acc_2) the previously accepted taxon (Acc_1) will now be linked to as a synonym (Syn_{2k}). As a consequence, Acc_1's links to higher taxa will simply be severed. However, facts, synonyms, and accepted included taxa previously linked directly to Acc1 are not connected to any potential taxon anymore and need to be further processed.

The taxonyms in the set of included taxa of lower rank (Acc_{11}, …, $Acc1_N$) will probably become synonyms of corresponding included taxa of Acc_2. The existence of the corresponding taxonyms in the system cannot be assumed, and even if present their identification is not trivial, so that an automation of this process is not possible. Instead, the status for all taxonyms in the now unlinked subtree of included taxa is set to ‘inactive’, so that they can be further processed as soon as the editor has identified or created the accepted taxon they will be linked to as a synonym.

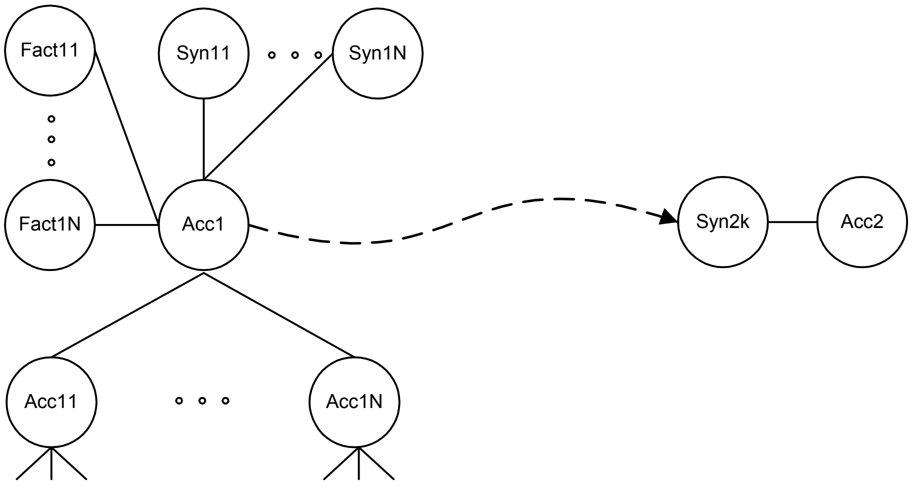


Figure 7: Status change from accepted to synonym

Factual data ($Fact_{11}$, … $Fact_{1N}$) linked directly to Acc_1 cannot stay linked to a synonym, so their potential taxon link has to be severed, too. Neither can they be transmitted automatically to the new accepted taxon (Acc_2), because conflicts with already existing facts may arise. Instead, the user has to decide on an individual basis whether and how a fact can be transferred. A set of forms has been designed for that purpose, where, for each fact category, the fact linked to Acc_1 can be compared with the corresponding facts of Acc_2. The user is able to merge the information by copying and pasting into a destination text field. Conflicts can be documented with a notes field.

Note that the original source of the information can be preserved in the system because the original status assignation and thus all relationships can be preserved with the original taxonym Acc_0. It is only when this has been copied and assigned to a different reference (author) that the need for status changes arises. An alternative way of transferring factual data is to state a congruent concept relationship between Acc_0 and Acc_2.

Finally, the user has to decide how to treat synonyms (Syn_{11}, …, Syn_{1N}) of the taxon to be altered since synonyms cannot be linked to other synonyms. For every synonym the following cases should be distinguished on an individual basis and a form should be designed implementing the respective choices:

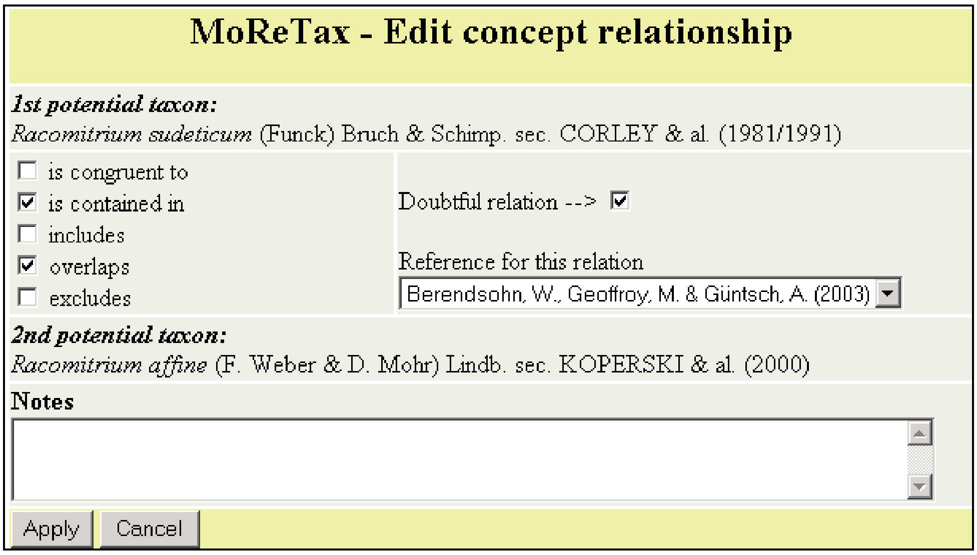
- The synonym will be an accepted taxonym of its own. In this case, the parent node of the new accepted taxonym has to be selected.
- The synonym becomes a synonym of the target accepted taxon (Acc_2).
- The synonym is marked as inactive to be resolved at a later stage. In this case, all relations to accepted names are severed, documented in the notes field of the potential taxon, and the status is changed from ‘synonym’ to ‘unresolved’. The latter change is necessary to maintain the status-dependent data integrity rule stating that a synonym must have a relation to an accepted name.

~~d)~~ The synonym will become a synonym of an unrelated accepted taxon. In this case, a quick search form has to support selecting the new accepted taxon.

Capturing concept relationships

A prominent feature of the MoReTax taxonomic editor will be the ability to enter and edit concept relationships between potential taxa as defined in the Berlin Model, thus enabling future information systems to propagate factual data through networks of concept relations and rate results as they are presented to users of such systems.

The concept editor provides a search form for selecting two taxonyms with accepted status from the database. Pre-selecting the names may either be based on the name or on the circumscription reference ("sec.").



MoReTax - Edit concept relationship

1st potential taxon:
Racomitrium sudeticum (Funck) Bruch & Schimp. sec. CORLEY & al. (1981/1991)

□ is congruent to
☑ is contained in
□ includes
☑ overlaps
□ excludes

Doubtful relation --> ☑

Reference for this relation
Berendsohn, W., Geoffroy, M. & Güntsch, A. (2003)

2nd potential taxon:
Racomitrium affine (F. Weber & D. Mohr) Lindb. sec. KOPERSKI & al. (2000)

Notes

Figure 8: Editing concept relationships

Once the two potential taxa involved are selected, the list of already existing relations between these concepts will be displayed and the user may choose to edit an existing relation or enter a new one. A form for this operation will have to provide the two potential taxa displayed and checkboxes to indicate a set of base relationships (see Figure 8 for an example form). Since users tend to confound the ordering of asymmetric relations (includes, is contained in) we recommend arranging the involved fields (first taxonym, relation, second taxonym) in a clear top-to-bottom sequence.

Additional fields are provided to indicate whether the relationship defined is considered doubtful and to select the reference for the relationship itself. This reference defaults to the author identified as current user of the editor. However, it should be possible to enter and edit concept relations assigned to other references (e.g. entering a concept relationship from a flora), too. A notes field will be used to capture remarks for the given relation.

Conclusion

This article summarises specifications discussed and compiled in the context of several parallel projects. MoReTax provided the results of a study analysing capabilities and constraints of existing taxonomic editors (GEOFFROY, unpublished) while the other projects all aim – at least in part – at the design and implementation of user interfaces capable of processing potential taxa and their related information.

This commonality set aside the projects pursue different objectives and timelines, which made the development of a common core a challenging task. Currently, the implementation of the World Wide Web editor used for the Euro+Med PlantBase project (EURO+MED, 2002) and for the Dendroflora of El Salvador (BERENDSOHN, in prep.) is nearing completion. This will serve as the base for a projected implementation of the taxonomic core functionality at the German Federal Agency for Nature Conservation, to be used to manage and remotely edit data holdings on the taxa occurring in Germany.

Simultaneously, desktop applications are being implemented to manage data for the AlgaTerra project (JAHN, 2002) and to support editing and publication of Volume 2 of the Med Checklist (GREUTER, unpublished). Both applications are Visual Basic (Access2000) clients linked to a database backend using the Berlin Model's taxonomic core. These efforts thus complement the web-based approach taken by Euro+Med and allow local management of their databases without the inherent disadvantages of the Web Editor.

In the process of application development, a toolkit will emerge that covers database management tasks not described in this article, such as user authorisations, correcting catalogue tables (e.g. geographic area names), importing and merging data sets, resolving duplicate records, and elimination of orphaned records.

Experience drawn from taxonomic work carried out by experts using the editor will be used to successively refine the full specification of the taxonomic editor software.

References cited

BERENDSOHN, W. G. (1997): A taxonomic information model for botanical databases: the IOPI model. Taxon 46: 283-309.

BERENDSOHN, W. G., DÖRING, M., GEOFFROY, M., GLÜCK, K., GÜNTSCH, A., HAHN, A., KUSBER, W.-H., LI, J.-L., RÖPERT, D. & SPECHT, F. (2003): The Berlin Taxonomic Information Model. Schriftenreihe Vegetationsk. 39: 15-42.

EURO+MED (2002): Introduction to the project. http://www.euromed.org.uk/a_bout/.

GEOFFROY, M. & GÜNTSCH, A (2001 [Jan 17 2003]): Handling factual information linked to parallel taxonomic concepts in biology. 17th meeting of the Taxonomic Databases Working Group, Sydney, Abstract Volume. http://plantnet.rbgsyd.gov.au/bioforum/TDWG_program/tdwg_abstracts.html.

GEOFFROY, M. & GÜNTSCH, A. (2003): Assembling and navigating the potential taxon graph. Schriftenreihe Vegetationsk. 39: 71-82.

GÜNTSCH, A., LI, J. & BERENDSOHN, W. (2002 [Dec 30]): Euro+Med PlantBase – Design of the Internet Taxonomic Sector Editor http://www.bgbm.org/BioDivInf/Projects/Euro+Med/EditorDesign.pdf.

JAHN, R. (2002 [20 Dec.]): AlgaTerra Information System. An information system for terrestrial algal biodiversity: a synthesis of taxonomic, molecular and ecological information. – http://www.algaterra.org/aims.htm.

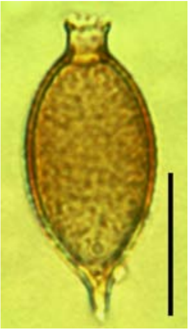
MACROMEDIA (2003 [Jan 17]): ColdFusion MX. http://www.macromedia.com/software/coldfusion/.

PANDO, F. (2000 [17 Jan 2003]): BIBMASTER: A database application for nomenclature, literature and specimen management http://www.rjb.csic.es/bibmaste/bibmaste.htm

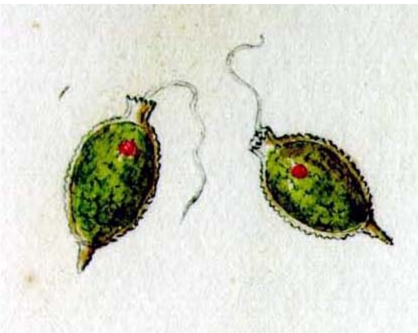
PANKHURST, R. J. (1993): Taxonomic Databases: The PANDORA System. In: FORTUNER R. (ed.): Advances in computer methods for systematic biology: Articificial Intelligence, databases, computer vision. John Hopkins University Press.

REVELATION (2002 [Dec 30]): Revelation software. http://www.revelation.com.

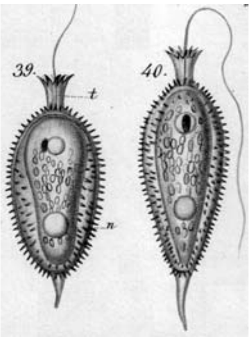
SUN (2003 [Jan 17]): JavaServer Pages – dynamically created web content. Sun Microsystems Inc. http://java.sun.com/products/jsp/.



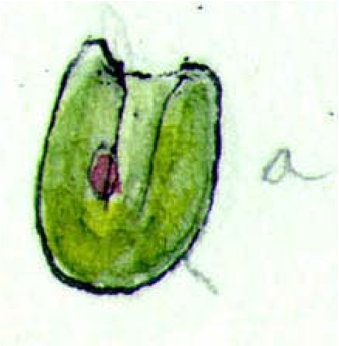
1



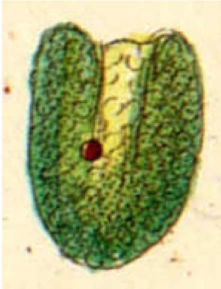
2



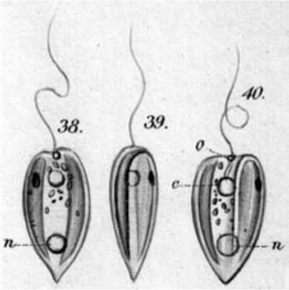
3



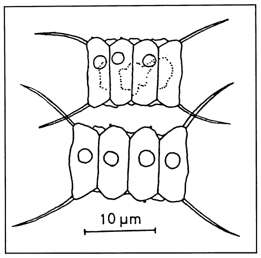
4



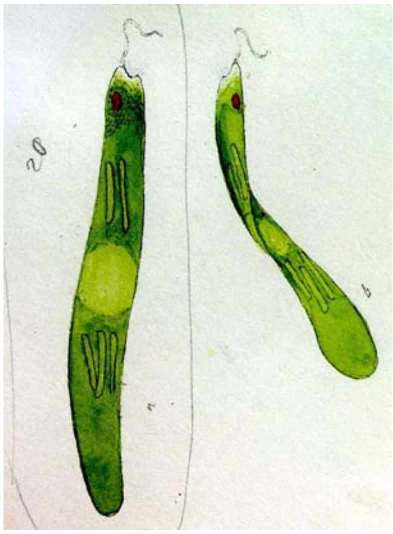
5



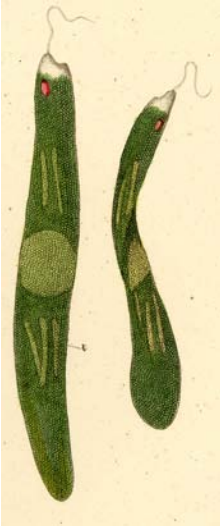
6



7



8



9

Typification - an extension of the Berlin Model

WOLF-HENNING KUSBER, KARL GLÜCK, MARC GEOFFROY & REGINE JAHN

AlgaTerra and the Berlin Model

Micro algae are one of the most diverse and at the same time comparatively unexplored groups of organisms. Because of their microscopic size, identification and the name-giving typification process is largely dependent on pictures or drawings, which cannot be verified as readily as types of larger organisms. This makes the group particularly suited for a concept-based information system, which accommodates both verified taxonomic and nomenclatural information and factual information connected to more or less clearly circumscribed potential taxa.

Several databases with information on algae have been set up over the last few years and made available on the Internet. Whereas the Index Nominum Genericorum is restricted to generic names and their type species (FARR & ZIJLSTRA, undated), the Diatom Genus Name Project (FOURTANIER & KOCIOLEK, 2001) and the Index Nominum Algarum (SILVA, 1997) include the reference of the type specimen and/or the reference of the picture. There are other databases that provide names, without type information but with some additional data, such as distribution data for algae or cyanobacteria (GUIRY & DHONNCHA, 2002; ANON., 2002a) or pictures and classifications (ANON., 2002b). However, the data is often not fully referenced and the information is linked by names only.

AlgaTerra, a project within the BIOLOG program of the German Federal Ministry of Education and Research, will provide fully referenced taxon, type, name, and collection data, images of types as well as ecological and molecular information on micro algae. Modelling, development, and implementation of the database as well as data evaluation are tasks of the project (JAHN, 2002).

The core of the Berlin Model (BERENDSOHN & al., 2003; BERENDSOHN & al., 2002) enables the AlgaTerra project to deal with a diversity of taxonomic concepts and their varying names from different sources. To fulfil the demands of an information system on micro algae, the Berlin Model is extended by the AlgaTerra project in two areas. On the one hand, tables with molecular, morphological and ecological data are attached to the table FACTS. Factual data associated with possibly different taxonomic concepts can be linked using the rules that are being developed in the MoReTax project (GEOFFROY & BERENDSOHN, 2003). On the other hand, the Berlin Model is complemented with the Type Designation Extension, involving several tables to assign a nomenclatural type to a name of a taxon and to document the verification of that information.

In this paper we explain nomenclatural types, give examples from the AlgaTerra project to illustrate the complexity and usefulness of the AlgaTerra approach (Examples 1-6, Figures 1-9), and describe the information model for the Type Designation Extension of the Berlin Model.

Nomenclatural types as a calibration tool

In order to use plants, especially micro algae, in biodiversity research and as indicators in biomonitoring and in palaeo-climate reconstruction it is necessary to rely on names of taxa. Unclear, imprecise, or unverifiable taxonomic concepts degrade the quality of information in systems based on names. Because concepts may and should evolve over time as part of the scientific process, true stability cannot and should not be aimed at. However, tools have to be developed to 'calibrate' the names used in these systems as much as is possible under the circumstances.

The only true consensus existing is the "calibration" given by the application of a set of rules to name taxa. These rules are laid down in the International Code of Botanical Nomenclature (GREUTER & al. 2000). The single most important principle of the Code is to connect each name with a nomenclatural type, that is, some concrete physical object (mostly a conserved sample of an organism) that can be re-examined to verify the application of a name.

Table 1: Type status, related information and examples for changes

Type status	Type Reference	Material	Change of status
Holotype	same as name reference	original	• can be replaced by a lectotype if the holotype is lost or destroyed • can be specified or superseded by a lectotype if it belongs to more than one taxon
Lectotype	other than name reference	original	• can be subsequently emended by a second-step lectotype
Neotype	other than name reference	other	• can be superseded by a lectotype if original material is found • can be subsequently emended by a second-step neotype
Epitype	same or other than name reference	any (see text)	• can be changed if the type to which it is linked is superseded by a different type
Any type	other than name reference	–	• can be corrected according to Art. 9.8 of the ICBN (GREUTER & al. 2000) if erroneously published (see Example 6)
Any type	ICBN	any material	• can be rejected or conserved by the ICBN after publication as a proposal in the journal *Taxon* (see Example 3)

There are a number of categories of nomenclatural types. Specimens which are part of the original material, that is, which have been in the hand of the author when proposing a new taxon name, may serve as holotype, syntypes, and/or paratypes, if they are cited or designated in the first publication (the protologue) of the name. A single specimen, explicitly designated as type in the reference of the first description, is the holotype. Otherwise, two or more specimens cited can be syntypes. Paratypes are specimens cited in the protologue but not designated as holotype or syntypes. A lectotype is selected at a later stage from original material if more precise information about the specimen is needed and/or if a holotype was not designated in the first description (see Examples 1-2). A neotype is designated from other material if no original material exists (see Example 4).

Each duplicate of a holotype is an isotype; each duplicate of other types is marked by the prefix "iso-", such as an isoneotype (see also Example 4). If a lecto- or a neotype specimen is in need of a more precise designation, a second-step lecto- or a second-step neotype can be selected (see Example 5). If the type specimen does not provide the needed characters for unambiguous taxonomic interpretation, an additional interpretative type, the epitype, can be designated. The epitype must be linked to the nomenclatural type.

In groups of microscopical plants, such as micro algae, which are only visible by microscope, illustrations play a relatively important role (Figures 1-9).

Since 1958 descriptions of non-fossil algae must be accompanied by an illustration; it is recommended that the specimen shown be the holotype (GREUTER & al. 2000, Art. 39). This illustration may serve as the type if the specimen cannot be preserved (Art. 37.4). Since 1912, descriptions of fossil algae (Art. 38) must be accompanied by an illustration; since 2001 this figure must show the type. In summary, a published (or unpublished) figure can be an illustration of a type or the type itself (see Example 6). Phycologists coined the term "iconotype" and used it for the last five decades, but it was never adopted by the Code for reasons of ambiguity: authors used it for both, illustrations of the type (icon of the type) and actual types (icon = type).

Examples for typified algal names from the AlgaTerra database project

We give six examples of micro algal types from the AlgaTerra project. Each example consists of one or more algal name(s) linked to the same type or types. Examples 1-3 are type designations, based on original material, published here for the first time. Example 4 deals with a second step typification; Example 5 shows problems that arise from conservation (or rejection) of types by the International Code of Botanical Nomenclature (GREUTER & al. 2000).

Example 1 (lectotype, original material), Figures 1-3

Chaetoglena caudata Ehrenb. in Ber. Bekanntm. Verh. Königl. Preuss. Akad. Wiss. Berlin 1840: p. 199. 1840.
Lectotype (designated here by W.-H. Kusber & R. Jahn): specimen shown in our Figure 1 from preparation "Trockenpräparate II Polygastrica" No. XXXII: 5 in the Ehrenberg Collection (BHUPM).
[Further original material, on which Ehrenberg's description was based: drawing No. 241 in the Ehrenberg Collection (BHUPM) shown here as Figure 2.]
[Comment on nomenclature: This name has been recombined as *Trachelomonas caudata* (Ehrenb.) F. Stein.]

Trachelomonas caudata (Ehrenb.) F. Stein, Organismus Infusionsthiere III, 1: legend to pl. 22: figs 39-40. 1878.
Basionym: *Chaetoglena caudata* Ehrenb. in Ber. Bekanntm. Verh. Königl. Preuss. Akad. Wiss. Berlin 1840: p. 199. 1840.
[Comment on nomenclature: Stein's combination is validly published and linked to the type of the name of the basionym.]
[Comment on taxonomy: *Trachelomonas caudata* (Ehrenb.) F. Stein sec. STEIN (1878), pl. 22: figs 39-40 (given here as Figure 3) is the taxonomic concept (potential taxon) taken over by later monographs, but the identity with Ehrenberg's taxonomic concept is doubtful.]

Example 2 (lectotype, unpublished figure, published figure), Figures 4-6

Cryptoglena pigra Ehrenb. in Abh. Königl. Akad. Wiss. Berlin 1831: p. 150. 1832.
Lectotype (designated here by W.-H. Kusber & R. Jahn): specimen marked with "B a" on drawing No. 353 in the Ehrenberg Collection (BHUPM). This figure given here in our Figure 4 is the original drawing for the illustration in EHRENBERG (1838) pl. 2: fig. 26, reproduced here as our Figure 5.
[Comment on taxonomy (1): Emendation by Ehrenb. in Abh. Königl. Akad. Wiss. Berlin 1833: p. 290, pl. 7: fig. 2. 1834.]
[Comment on taxonomy (2): *Cryptoglena pigra* Ehrenb. sec. STEIN, Organismus Infusionsthiere III, 1: pl. 19: figs 38-40 (reproduced here as our Figure 6) is the taxonomic concept (potential taxon) taken over by later monographs, but the identity with Ehrenberg's taxonomic concept is doubtful.]

Example 3 (conserved type, provisional conserved type), Figure 7

Achnanthes quadricauda Turpin in Mém. Mus. Hist. Nat. 16: p. 311. 1828.
Conserved type according to GREUTER & al. (2000), p. 376: "[Specimen from strain] Hungary, Lake Belsö-tó, Hegewald 1971/256 (Kernforschungsanlage Jülich, Germany)".
[Comment on nomenclature: The basis for conservation was the "typ. cons. prop." i.e. the provisional conserved type by Compère & Komárek in Taxon 39: p. 530. 1990.]
[Comment on taxonomy: the upper cell of our Figure 7, reproduced from HEGEWALD (1977) is the "type fig." i.e. the icon of the type.]

Scenedesmus quadricauda (Turpin) Bréb. in Brébisson, L. A. & Godey, L. L.: Alg. Falaise: p. 66. 1835.
Basionym: *Achnanthes quadricauda* Turpin in Mém. Mus. Hist. Nat. 16: p. 311. 1828.
≡ *Scenedesmus communis* E. H. Hegew., nom. illeg. in Arch. Hydrobiol. Suppl. 51: p. 151, figs 12-13. 1977.
≡ *Desmodesmus communis* (E. H. Hegew.) E. H. Hegew., nom. illeg. in Arch. Hydrobiol. Suppl. 131: p. 8. 2000.
[Comment: for details on further taxonomic implications see KUSBER & JAHN (2002).]

Figures 1-9 (facing page)

Figures 1-2: *Chaetoglena caudata* Ehrenb. Figure 1: lectotype (prep. No. XXXII:5, BHUPM), scale bar = 20 µm. Figure 2: original material (drawing No. 241, BHUPM), cell length = 31.3 µm.

Figure 3: *Trachelomonas caudata* (Ehrenb.) F. Stein sec. STEIN (1878: pl. 22: figs 39-40).

Figures 4-5: *Cryptoglena pigra* Ehrenb. Figure 4: lectotype (specimen "B a" on drawing No. 353, BHUPM), cell length = 9.0 µm. Figure 5: Icon of the lectotype, reproduced from EHRENBERG (1838: pl. 2: fig. 26).

Figure 6: *Cryptoglena pigra* Ehrenb. sec. STEIN (1878: pl. 19: figs 38-40).

Figure 7: *Achnanthes quadricauda* Turpin, upper cell is icon of the conserved type, reproduced from HEGEWALD (1977: figs 12-13, as icon of the type for *Scenedesmus communis* E. H. Hegew., nom. illeg.) with kind permission by E. Hegewald.

Figures 8-9 : *Amblyophis viridis* Ehrenb. Figure 8a: neotype (fig. "a" from drawing No. 77, BHUPM), cell length = 225.6 µm. Figure 8b: isoneotype (fig. "b" from drawing No. 77, BHUPM). Figure 9: icons of the neotype (left) and isoneotype (right), reproduced from EHRENBERG (1838, pl. 7: fig. 5).

INSERT Figures (colour plate) here

Example 4 (neotype, isoneotype), Figures 8-9.

Amblyophis viridis Ehrenb. in Abh. Königl. Akad. Wiss. Berlin 1831: p. 73. 1832.
= *Amblyophis viridis* Ehrenb. emend. Ehrenb. in Abh. Königl. Akad. Wiss. Berlin 1835: p. 165: fig 17. 1836.
Neotype (designated here by W.-H. Kusber & R. Jahn): marked specimen shown in fig.“a” from drawing No. 77 in the Ehrenberg Collection (BHUPM), Ehrenberg’s figure reproduced here as Figure 8a.
Isoneotype (designated here by W.-H. Kusber & R. Jahn): specimen shown in fig.“b” from drawing No. 77 in the Ehrenberg Collection (BHUPM), Ehrenberg’s figure reproduced here as Figure 8b.
[Comment on taxonomy and nomenclature: Apart from the lack of material the drawing is signed as “1835”, the drawing No. 77 is the first illustration of *A. viridis*, published in Ehrenberg’s emendation of the first diagnosis of *A. viridis* in 1832. The illustration was again published in EHRENBERG (1838) as pl. 7: fig. 5, reproduced here in part as our Figure 9.]
[Comment on nomenclature: This name has been recombined and renamed as *Euglena ehrenbergii* Klebs.]

Euglena ehrenbergii Klebs in Untersuch. Bot. Inst. Tübingen 1(2): p. 304, pl. 2: figs 1-3, 5. 1883.
Replaced synonym: *Amblyophis viridis* Ehrenb. in Abh. Königl. Akad. Wiss. Berlin 1831: p. 73. 1832. (cited erroneously as: “Ehbg. S. 103. Taf. VII, Fig. 5” i.e. *Amblyophis viridis* Ehrenb. in Infusionsthierchen: p. 103: pl. 7: fig. 5).
[Comment on taxonomy: Apart from the formal replacement of Ehrenberg’s name, KLEBS (1883) emended Ehrenberg’s taxon concept (EHRENBERG 1838) and provided more detailed figures concerning the apex of the cell.]

Example 5 (lectotype, second-step lectotype, second-step isolectotype).

Navicula pupula Kütz., Kieselschal. Bacill., p. 93, pl. 30: fig. 40. 1844.
Lectotype: BM slide 17918, designated by R. Ross in Bull. Brit. Mus. (Nat. Hist.), Bot 3 (2), p. 87. 1963.
Second step lectotype: specimen on BM slide 17918 (England Finder M45/2), designated by D. G. Mann in R. Jahn et al., Lange-Bertalot-Festschrift, p. 236. 2001.
Second-step isolectotypes: four specimens (BM slide 17918: England Finder L39/1-3, M37/2, S35/2-R35/4, J32/2-4), designated by D. G. Mann in R. Jahn et al., Lange-Bertalot-Festschrift, p. 236. 2001.
Comment: The second step lectotype specimen on BM slide 17918 (Finder M45/2) is shown in MANN (2001) on figs 2-6. The second-step isolectotype specimens are shown in MANN (2001) on figs 7-10 (= Finder L39/1-3), figs 11-12 (= Finder M37/2), figs 13-14 (= Finder S35/2-R35/4), and figs 15-16 (= Finder J32/2-4).
[Comment on nomenclature: This name has been recombined as *Sellaphora pupula* (Kütz.) Mereschk.]

Sellaphora pupula (Kütz.) Mereschk. in Ann. Mag. Nat. Hist., ser. 7, 9: p. 187. 1902.
Basionym: *Navicula pupula* Kütz. in Kieselschal. Bacill., p. 93, pl. 30: fig. 40. 1844.

Example 6 ("iconotype", holotype)

Euglena smulkowskiana Zakryś in Nova Hedwigia 42: p. 524, pl. 4: fig. 6. 1986.
Iconotype: Nova Hedwigia 42: pl. 4: fig. 6. 1986.
[Comments on nomenclature: The term "iconotype", not applicable according to the ICBN (GREUTER & al. 2000, Art. 9.8) has to be corrected into "holotype". This name has been recombined as *Phacus smulkowskianus* (Zakryś) Kusber.]

Phacus smulkowskianus (Zakryś) Kusber in Willdenowia 28: p. 246. 1998.
Basionym: *Euglena smulkowskiana* Zakryś in Nova Hedwigia 42: p. 524, pl. 4: fig. 6. 1986.
[– *Phacus similis* Christen, nom. inval. in Rev. Algol. 6: p. 164, 195, pl. 1: fig 3-4. 1962.]
[Comments on taxonomy: The potential taxon *Phacus similis* Christen sec. CHRISTEN (1962) has been evaluated as being congruent to *Euglena smulkowskiana* Zakryś sec. ZAKRYŚ (1986) by KUSBER (1998, p. 246). Ecological data and documentation, published in KUSBER (1998) is linked to the potential taxon *Phacus smulkowskianus* (Zakryś) Kusber sec. KUSBER (1998) which includes both potential taxa as well as *Phacus similis* f. *minor* Bourr. & Couté, nom. inval. (see KUSBER, 1998).]

The Type Designation Extension of the Berlin Model

The examples given should suffice to demonstrate the intricacy of the taxonomic and nomenclatural process and its tight interrelation with type designation events. The extension to the Berlin Model needed by AlgaTerra (and comparable projects) mirrors this complexity (Figure 10). The extension's prime relationship with the core model is the NAME table's link to the TYPEDESIGNATION table. Apart from that, the core bibliographic system is used via the REFDETAIL table, either indirectly (nomenclatural reference of the name) or directly (various sources and bibliographic references). In all, 10 new tables are defined in the extension, to which the type designation event is central (table TYPEDESIGNATION). As discussed above, the handling of specimens differs markedly from that of illustrations. Therefore, two tables were defined to handle their linkage with the central event (TYPEFIGUREDESIGNATION and TYPESPECIMEN-DESIGNATION).

Methods and conventions follow those defined in BERENDSOHN & al. (2003). Every table in the Type Extension has an attribute for notes, which was omitted in the tables as shown below.

The designation event

Clusters of taxon names can be traced to one name actually connected to the type specimen. This name is the basionym or the replaced synonym of a name. The core model covers such relationships between names, so the underlying type for every name can be retrieved using the NAME and RELNAME tables.

Every designation event documented in the table TYPEDESIGNATION (Table 2) refers to exactly one name, but more than one specimen and/or picture can be designated as a type in one event. In AlgaTerra, experts evaluate all designation events in the system, after scrutiny of the original material, where possible.

The flag in the ImplicitExplicit attribute of the TYPEDESIGNATION table enables the expert to show whether the designation in a given reference had been done explicitly (e.g. “designated here”) or implicitly by only mentioning that a specimen is the type of a given name (allowed only until the year 2000).

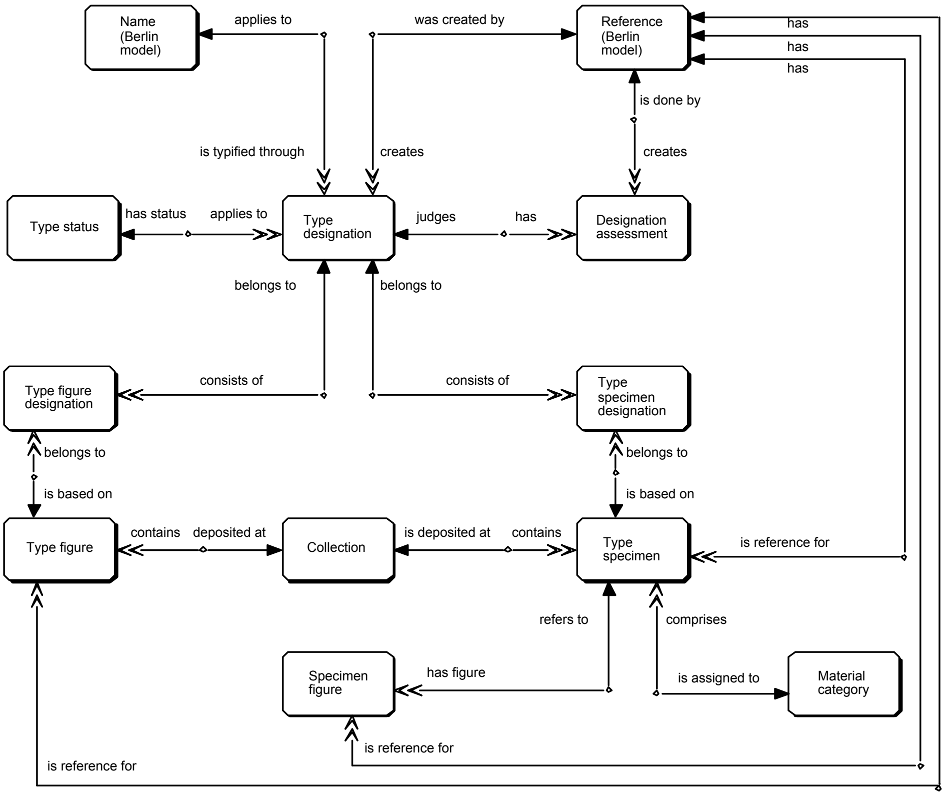


Figure 10: ER Diagram for the Type Extension

In any case a type designation assigns an object to a name of a taxon. To ease information retrieval, the TypeCache attribute contains the complete type citation string necessary for a valid publication. In our database it is composed by a trigger, which refreshes the string on updates or inserts in the attached tables. The TypeStatusFk attribute assigns a Status of the TYPESTATUS catalogue to the type designation.

Type status

The table TYPESTATUS contains only two string attributes apart from its primary key: InformalStatus and Status. For the latter, the following values are currently defined: ‘epitype’, ‘holotype’, ‘isolectotype’, ‘isoneotype’, ‘isotype’, ‘lectotype’, ‘neotype’, ‘paraneotype’, ‘paratype’, ‘second-step lectotype’, ‘second-step neotype’, ‘syntype’, ‘iconotype’. The InformalStatus field in the TYPESTATUS table indicates a status that is not applicable according to the ICBN, for example, an “iconotype” or a “phototype”.

Table 2: Attributes of table TYPEDESIGNATION attributes

Short name	Type	Description
TypeDesignationId	int	Primary key of table TYPEDESIGNATION
NameFk	int	Pointer to NAME
TypeStatusFk	int	Pointer to TYPESTATUS
TypeCache	str	The complete type designation as to be cited
RefFk	int	Pointer to REFDETAIL (combined foreign key), indicating the (published) source of the designation
RefDetailFk	int	
ImplicitExplicit	int	Indication whether the type is implicitly or explicitly designated

Assessment of a designation

A name in the NAME table may be linked to more than one designation. To evaluate a given designation the table DESIGNATIONASSESSMENT (Table 3) contains the fields to indicate which of several designations are the ones preferred by a certain named source.

The ConsRejFlag attribute indicates whether a type is conserved (see Example 3) or rejected by the Nomenclatural Session of the International Botanical Congress (GREUTER & al. 2000, Appendix III). The PrefBySystemFlag is set in the DESIGNATIONASSESSMENT table on the designation preferred by the person or group managing the database system. The attributes VerifiedBy and VerifiedWhen denote the who and when of the authorisation of the data, while the pointer to the REFDETAIL table may denote a reference to an expert designation assessment. The PrefBySystemFlag allows to give users the option to restrict output to expert evaluated type information related to the name.

Table 3: Attributes of table DESIGNATIONASSESSMENT

Short name	Type	Description
DesignationAssessmentId	int	Primary key of DESIGNATIONASSESSMENT
TypeDesignationFk	int	Pointer to TYPEDESIGNATION
PrefBySystemFlag	bool	Flag to indicate the preferred type designation
VerifiedBy	str	Name of expert who verified the assessment
VerifiedWhen	date	Date when the expert verified the assessment
PrefByRefFk	int	Pointer to REFDETAIL (combined foreign key), indicating the printed reference for an expert's verification
PrefByRefDetailFk	int	
ConsRej	str	Indication whether the type is conserved or rejected by the ICBN
ConsRejRefFk	int	Pointer to REFDETAIL (combined foreign key), indicating the printed reference for the conservation or rejection (or its proposal)
ConsRejRefDetailFk	int	

Pictures and specimens as the basis for type designation

The two objects in the physical world that stand for the type of a name of a taxon are the type specimen itself (specimen, shown in Figure 1) or the illustration of the specimen (until now unpublished Figures: 4, 8, earlier published Figures: 5, 7, 9). Both specimens and unpublished figures are frequently parts of a collection (see drawings in Example 1-2, 4). Data on these physical objects are stored in our database in two different tables, TYPEFIGURE and TYPESPECIMEN. Both are linked to COLLECTION (Table 4) holding information on the collection itself, e.g. the name and the town of the collection in the attributes Name and Town, the international herbarium code in the attribute IHCode (according to Index Herbariorum, HOLMGREN & HOLMGREN, 2002) and, where applicable, information on collections consisting of distinguishable, named collections in the attribute Subcollection (e.g. Ehrenberg Collection at BHUPM, see Examples 1, 2, 4).

Table 4: Attributes of table COLLECTION

Short name	Type	Description
CollectionId	int	Primary key of table COLLECTION
Name	str	Name of the herbarium
Town	str	Town where the herbarium is located
IHCode	str	International herbarium code
Subcollection	str	Name of a collection in the named herbarium above

Although a specimen is generally designated as type, both a picture (TYPEFIGURE) and a specimen (TYPESPECIMEN) or only one of them can be used to assign a type to a name. The many-to-many relationship between the respective tables and the TYPEDESIGNATION table is resolved in the tables TYPEFIGUREDESIGNATION and TYPESPECIMENDESIGNATION.

One name may have several type designations with different status, which can be identified by their references (Example 5).

Pictures

If the name is typified by a single photograph or drawing only (see Example 6) and the designation has been called “iconotype” this information is inserted into the InformalStatus attribute of the table TYPESTATUS. In this example the Status itself is ‘holotype’. In the TypeFigure attribute of TYPEFIGURE (Table 5) the picture itself is represented as a link to the file of the picture. A picture that has been designated as a type may have been published in a source different from the designation. The TYPEFIGURE table therefore has a link to the REFERENCE section. Further information on the picture, such as its legend derived from the reference of the figure, is stored in the TypeFigurePhrase attribute. This attribute may also contain information on the type locality and the collector, in the case it is the type figure (see Table 5). The IsTypeFlag is set to mark a picture directly reproduced from the type (our Figures 4, 8). It is not set if the system shows a reproduction of the original type figure, which can differ more or less from the type (for comparison see Examples 2, 4 and our Figures 5, 9).

Table 5: Attributes of table TYPEFIGURE

Short name	Type	Description
TypeFigureId	int	Primary key for table TYPEFIGURE
TypeFigurePhrase	str	Indication of the type figure or figure of the type
TypeFigure	str	Pointer to the file showing the type figure
RefFk	int	Pointer to REFDETAIL (combined foreign key), indicating the place of publication of the picture
RefDetailFk	int	
IsTypeFlag	bool	Indication if the figure is the type
IsPublished	bool	Figure is published or not
CollectionFk	int	Pointer to COLLECTION

Pictures not typified but showing a typified specimen used in a type designation are stored in the SPECIMENFIGURE table (Table 9; Figure 1).

Specimens

In the case of algal names type vouchers may even be living material, if preserved in a metabolically inactive state, according to the rules of botanical nomenclature (GREUTER & al. 2000, Art. 8.4). To encompass all possible materials of a type specimen the catalogue table MATERIALCATEGORY is attached to the TYPESPECIMEN table. It contains (apart from its primary key) the string attribute MaterialCategory, currently with the values: 'culture' (permanently preserved), 'fossil', 'herbarium sheet', 'published figure', 'sample', 'microscopic slide', 'unpublished figure', 'wet preparation'.

In the table TYPESPECIMEN (Table 6) information specific to the specimen itself can be found, such as the type locality, the collector and the precise location on the microscopic slide (finder number). As the specimen may be published in a source differing from that of the designation it is linked to the REFDETAIL table as well.

Table 6: Attributes of table TYPESPECIMEN

Short name	Type	Description
TypeSpecimenId	int	Primary key for table TYPESPECIMEN
TypeSpecimenCache	str	Complete type specimen citation string
TypeSpecimenPhrase	str	Indication of the type specimen as to be cited
TypeLocality	str	Indication of the type locality as to be cited
CollectionFk	int	Pointer to COLLECTION
RefFk	int	Pointer to REFDETAIL (combined foreign key), indicating another publication of the specimen
RefDetailFk	int	
MaterialCategoryFk	int	Pointer to MATERIALCATEGORY

The botanical name may be recorded with more than one type status in one reference (see Example 4) or the same specimen may be mentioned with differing type status in different references (see Table 1 and Example 6). In the case of a preparation containing several individuals (e.g. microscopic slide) the entire preparation (GREUTER & al. 2000, Art. 8.2) as well as marked individuals can be used as specimen for typification

(Example 5, see discussion in MANN, 2001). The tables TYPESPECIMEN and TYPEDESIGNATION are therefore linked with many-to-many relations.

Pictures of specimens used in designations

The link from the table TYPESPECIMEN to the table SPECIMENFIGURE enables us to attach one or more pictures to one specimen in a type designation. Pictures stored in this table are not types themselves (stored in TYPEFIGURE; see also Figure 7, Example 3), but they are figures of material used in the context of a type designation to visualise the typified specimen. If the picture shows more than one specimen or a subset of the typified specimen, the SubOrSuperSetFlag is set and the details are described in the SubOrSuperSetPhrase attribute. Pointers to the reference section of the model denote the source of a published picture of a specimen

Table 7: Attributes of table SPECIMENFIGURE

Short name	Type	Description
SpecimenFigureId	int	Primary key of table SPECIMENFIGURE
TypeSpecimenFk	int	Pointer to TYPESPECIMEN
SpecimenFigurePhrase	str	Indication of the figure of the specimen
SpecimenFigure	str	Pointer to the file showing the specimen figure (may be a URL)
SubOrSuperSetFlag	bool	Indication if the figure shows a part of the specimen or several specimens
SubOrSuperSetPhrase	str	Detailed description of the shown part or the (location within) the set of specimens
PrefFigureFlag	bool	Flag to indicate the preferred figure of the specimen
VerifiedBy	str	Name of expert who verified the figure of the specimen
VerifiedWhen	date	Date when the expert verified the figure of the specimen
RefFk	int	Pointer to REFDETAIL (combined foreign key), indicating the (published) source of the picture
RefDetailFk	int	

The attributes VerifiedBy and VerifiedWhen denote the who and when of the authorisation of the link between the picture of the specimen and the specimen and lead to the preferred view indicated by the value in the PrefFigureFlag attribute.

Accessing the type information

Given the choice, users wanting to identify taxa will preferably make use of the availability of the ConsRejFlag attribute and the PrefBySystemFlag attribute in the DESIGNATIONASSESSMENT table to restrict their access to include only the verified type information. However, users with a wider taxonomic interest may reproduce the methods the expert has deployed when he created or altered a type designation. Rejected or altered designations and their references are stored in the database as well.

By scrolling through the precursors of preferred or conserved type designations and their references users will be able to compare them with type designations of other names and to clarify their linked taxonomic concepts.

The information system including the Type Designation Module will be made available in the Internet for all users (see JAHN, 2002).

Perspective

Within the ongoing AlgaTerra project the Type Designation Extension will be tested and optimised in order to become part of the core of the Berlin Taxonomic Information Model. The extension will enable storing and updating of fully referenced type-related taxonomic knowledge from different sources in order to overcome nomenclatural and taxonomic shortcomings. Thus it will preserve, pool, and publish expert knowledge from over 200 years of taxonomic research.

Type Information from important German Collections (Ehrenberg Collection at BHUPM, Hustedt Collection at BRM, Lange-Bertalot Collection) will cover a good part of common freshwater diatoms. Besides the Type Designation Extension, AlgaTerra will use the core structure of the Berlin Taxonomic Information Model for linking molecular and ecological factual data, provided by the Alfred-Wegener Institut (Bremerhaven, Germany), Universität Leipzig, and SAG (Universität Göttingen), to potential taxa of micro algae. Rules on data integrity will be implemented in the AlgaTerra database. These integrity rules can also be used to test data of taxonomic research, e.g. citing, differentiating, and classifying synonyms.

The gathering and circulating of evaluated information through AlgaTerra will offer a firm basis for different demands and will give an impetus to taxonomic research as well as applied science.

This information is also needed in efforts such as the Global Biodiversity Information Facility, where biodiversity data needs to be linked to verified names – which depend on evaluated type data.

Acknowledgements

We would like to thank David Lazarus (Museum für Naturkunde, Berlin) for access to the Ehrenberg Collection and his comments on the manuscript.

References

ANONYMOUS (2002a [7 Feb 2003]): BIOS - Bacteriology Insight Orienting System. http://www-sp2000ao.nies.go.jp/cgi-bin/query.cgi?type=3.

ANONYMOUS (2002b [14 Jan 2003]): micro*scope. http://www.mbl.edu/microscope.

BERENDSOHN, W. G., GEOFFROY, M., GÜNTSCH, A. & LI, J.-J. (2002 [30 Dec.]): The Berlin Taxonomic Information Model. http://www.bgbm.org/biodivinf/docs/bgbm-model/.

BERENDSOHN, W. G., DÖRING, M., GEOFFROY, M., GLÜCK, K., GÜNTSCH, A., HAHN, A., KUSBER, W.-H., LI, J.-J., RÖPERT, D. & SPECHT, F. (2003): The Berlin Taxonomic Information Model. Schriftenreihe Vegetationsk. 39: 15-42.

CHRISTEN, H. R. (1962): Neue und wenig bekannte Eugleninen und Volvocalen. Rev. Algol. 6: 162-202.

EHRENBERG, C. G. (1838): Die Infusionsthierchen als vollkommene Organismen. Leipzig.

FARR, E. & ZIJLSTRA, G. (undated [14 Jan 2003]): Index Nominum Genericorum (Plantarum). http://rathbun.si.edu/botany/ing/.

FOURTANIER, E. & KOCIOLEK, P. (2001 [14 Jan 2003]): Diatom Genus Project. http://www.calacademy.org/research/diatoms/genproject/index.html.

GEOFFROY, M. & BERENDSOHN, W. G. (2003): The concept problem in taxonomy: importance, components, approaches. Schriftenreihe Vegetationsk. 39: 5-14.

GREUTER, W., MCNEILL, J., BARRIE, F. R., BURDET, H. M., DEMOULIN, V., FILGUEIRAS, T. S., NICOLSON, D. H., SILVA, P. C., SKOG, J. E., TREHANE, P., TURLAND, N. J., HAWKSWORTH, D. L. (2000): International Code of Botanical Nomenclature (Saint Louis Code) adopted by the Sixteenth International Botanical Congress St. Louis, Missouri, July - August 1999. Regnum Veg. 138: XVIII, 1-474.

GUIRY, M. D. & DHONNCHA, E. (2002 [14 Jan 2003]): AlgaeBase. http://www.algaebase.org.

HEGEWALD, E. (1977): *Scenedesmus communis* Hegewald, a new species and its relation to *Scenedesmus quadricauda* (Turp.) Bréb. Arch. Hydrobiol. Suppl. 51: 142-155.

HOLMGREN, P. & HOLMGREN, N. (2003 [Feb 3]): Index Herbariorum, The Herbaria of the World. New York Botanical Garden. http://www.nybg.org/bsci/ih/ih.html

JAHN, R. (2002 [20 Dec.]): AlgaTerra Information System. An information system for terrestrial algal biodiversity: a synthesis of taxonomic, molecular and ecological information. http://www.algaterra.org/aims.htm.

KLEBS, G. (1883): Über die Organisation einiger Flagellaten-Gruppen und ihre Beziehungen zu Algen und Infusorien. Untersuch. Bot. Inst. Tübingen 1: 233-361, Taf. 2-3.

KUSBER, W.-H. (1998): A study on *Phacus smulkowskianus* (Euglenophyceae) - a rarely reported taxon found in waters of the Botanic Garden Berlin-Dahlem. Willdenowia 28: 239-247.

KUSBER, W.-H. & JAHN, R. (2002): Standards für die Artidentifikation in der Limnologischen Forschung. Pp. 858-863. In: DEUTSCHE GESELLSCHAFT FÜR LIMNOLOGIE [ed.]: Jahrestagung 2001 (Kiel). Tutzing.

MANN, D.G. (2001): The systematics of the *Sellaphora pupula* complex: typification of *S. pupula*. – In: JAHN, R., KOCIOLEK, J. P. WITKOWSKI, A. & COMPÈRE, P. [ed.]: Lange-Bertalot-Festschrift: 225-241. Gantner, Ruggell.

SILVA, P. C. (1997 [14 Jan 2003]): Index Nominum Algarum, University Herbarium, University of California, Berkeley. Compiled by Paul Silva. http://ucjeps.berkeley.edu/INA.html.

STEIN, F. (1878): Der Organismus der Infusionsthiere. 3 (1). Leipzig.

ZAKRYŚ, B. (1986): Contribution to the monograph of Polish members of the genus *Euglena* Ehrenberg 1830. Nova Hedwigia 42: 491-540.

Assembling and navigating the potential taxon graph

MARC GEOFFROY & ANTON GÜNTSCH

Understanding the relationships between potential taxa is an essential pre-requisite for the construction of a reliable information access system based on potential taxon names. The relationship influences the way in which information linked to one name can be transferred or combined with information linked to another name. This process of aggregation or combination of factual information, here called transmission of linked information, is a fundamental requirement of users of taxonomic information systems.

If an organism name is used in a reference then a taxonym arises. The potential taxon thus named adheres to the explicit or implicit criteria expressed to denote the taxonomic concept and to draw the boundary between itself and the other potential taxa within that reference (see GEOFFROY & BERENDSOHN, 2003, for a definition of the terms taxonym, potential taxon, and taxon concept). A taxonomic concept is therefore a "subjective" view of a taxon. The “subjective” circumscription of a taxon constitutes the taxon concept. It should be noted that different taxonyms may lead to identical potential taxa but that the other way around different potential taxa cannot be identified by the same taxonym. In the language of mathematics we can assert that the relation between scientific names and potential taxa is just that of a relationship while the relation between taxonyms and potential taxa is also a function. This fact alone denotes an enormous advantage of taxonyms over scientific names.

For further discussion of the complexities arising from this seemingly simple notion we use an example from the Checklist of German Mosses (KOPERSKI & al., 2000). The authors have their own taxonomic concept for the species *Racomitrium affine* (F. Weber & D. Mohr) Lindb. within the Bryophyta which sometimes differs from the taxonomic concepts of others authors (box 1).

IF
Racomitrium affine (F. Weber & D. Mohr) Lindb. sec. KOPERSKI & al. (2000) is designated as PT_1,
Racomitrium affine (F. Weber & D. Mohr) Lindb. sec. FRAHM & FREY (1992) as PT_2,
Racomitrium affine (F. Weber & D. Mohr) Lindb. sec. LUDWIG & al. (1996) as PT_3,
Racomitrium heterostichum (Hedw.) Brid. sec. SMITH (1980) as PT_4,
Rhacomitrium heterostichum var. *affine* (Schleich.) J. J. Amann sec. MÖNKEMEYER (1927) as PT_5,
Rhacomitrium heterostichum var. *gracilescens* Bruch & Schimp. sec. MÖNKEMEYER (1927) as PT_6,
Racomitrium affine (F. Weber & D. Mohr) Lindb. sec. CORLEY & al. (1981/1991) as PT_7,
Racomitrium sudeticum (Funck) Bruch & Schimp. sec. CORLEY & al. (1981/1991) as PT_8, and
Racomitrium affine (F. Weber & D. Mohr) Lindb. sec. SMITH (1980) as PT_9,

THEN
the basic relationships which have been explicitly established by KOPERSKI & al. (2000) are:

$PT_1 \equiv PT_2$, $PT_1 \equiv PT_3$, $PT_1 \subset PT_4$, $PT_1 \supset PT_5$, $PT_1 \supset PT_6$, $PT_1 \supset PT_7$, $PT_1 \oplus PT_8$, $PT_1 ! PT_9$.

Box 1: Example data from KOPERSKI & al. (2000) and their representation as relationships between potential taxa. The set notation is explained in GEOFFROY & BERENDSOHN (2003).

Combining relationships

One of the basic relationships discussed in the beginning of this volume (Geoffroy & Berendsohn, 2003) applies to the relationship between the constituents of every conceivable pair of potential taxa. Within one and the same source reference (e.g. a taxonomic monograph), these relationships are mostly implied; e.g. there should be an exclusive relationship between any two taxa designated by accepted names of the same rank within the same treatment. However, once potential taxa from different sources are compared, the possibilities to automatically deduce concept relationships from the taxonyms involved are severely restricted. In most cases, expert knowledge is required to interpret such relationships. Although this process is part of any taxonomist's revisionary work, we cannot realistically expect experts to establish relationships for any conceivable pair of potential taxa (even if the task was restricted to those with useful linked information attached to them).

Therefore the question arises how to calculate the relationship between potential taxon PT_1 and PT_3 if only the relationships of each of these to a third one (PT_2) are known. For example, assuming that $PT_1 \supset PT_2$ and $PT_3 \oplus PT_2$, what can be said about the relationship between PT_1 and PT_3? The answer is not straightforward: the relationship can be either $PT_1 \supset PT_3$ or $PT_1 \oplus PT_3$, i.e. we must allow for several possible basic relationships between potential taxa. This is one reason for introducing the notion of "combined relationships". The other reason is that experts may not have sufficient information to establish a precise basic relationship between two potential taxa, and in consequence name several possibilities.

Combined relationships

Call R the set of all basic relationships. $R = \{\equiv, \subset, \supset, \oplus, !\}$. Every subset S_x of R describes a "combined relationship". This means that if PT_1 is related to PT_2 through the combined relationship S_x then one of the basic relationships that belong to (the subset) S_x is the basic relationship between PT_1 and PT_2.

$$(PT_1\ S_x\ PT_2) \Rightarrow \exists\ R_i \in S_x \mid (PT_1\ R_i\ PT_2)$$

There are 32 (2^5) different combined relationships:

$\emptyset$, $\{\equiv\}$, $\{\subset\}$, $\{\supset\}$, $\{\oplus\}$, $\{!\}$, $\{\equiv, \subset\}$, $\{\equiv, \supset\}$, $\{\equiv, \oplus\}$, $\{\equiv, !\}$, $\{\subset, \supset\}$, $\{\subset, \oplus\}$, $\{\subset, !\}$, $\{\supset, \oplus\}$, $\{\supset, !\}$, $\{\oplus, !\}$, $\{\equiv, \subset, \supset\}$, $\{\equiv, \subset, \oplus\}$, $\{\equiv, \subset, !\}$, $\{\equiv, \supset, \oplus\}$, $\{\equiv, \supset, !\}$, $\{\equiv, \oplus, !\}$, $\{\subset, \supset, \oplus\}$, $\{\subset, \supset, !\}$, $\{\subset, \oplus, !\}$, $\{\supset, \oplus, !\}$, $\{\equiv, \subset, \supset, \oplus\}$, $\{\equiv, \subset, \supset, !\}$, $\{\equiv, \subset, \oplus, !\}$, $\{\equiv, \supset, \oplus, !\}$, $\{\subset, \supset, \oplus, !\}$, and $\{\equiv, \subset, \supset, \oplus, !\}$.

Note that it would be an error to treat the empty relationship ($\emptyset$) as being equal to R5 (!). The empty relationship can only be used for the representation of a logical contradiction when operating with relationships. The assertion that two concepts exclude each other (!) is not a logical contradiction!

The combined relationship R (i.e. the set of all basic relationships) describes the fact that every basic relationship could occur; this is the same as to say that nothing is known about the real relationship between two taxonomic concepts.

Relationship qualifier

An expert may express some doubt about a relationship between two potential taxa. To take this into account any relationship may be flagged as "doubtful" ("?"). So the new expressions

$$\equiv?,\ \subset?,\ \supset?,\ \oplus?,\ !?\ \text{and}\ S?$$

are meaningful.

In the example mentioned above the authors have qualified two relationships as doubtful, that for *Racomitrium heterostichum* (Hedw.) Brid. sec. SMITH (1980) and that for *Racomitrium affine* (F. Weber & D. Mohr) Lindb. sec. CORLEY & al. (1981/1991). So these relationship should be described as $PT_1 \subset? PT_4$ and $PT_1 \supset? PT_7$.

The concatenation

We can now describe more precisely what happens if there is a relationship S_1 between PT_1 and PT_2 and another relationship S_2 between PT_2 and PT_3. For this purpose a "concatenation" operator "→" between two combined relationships can be defined, the result of which is another combined relationship:

$S_1 \rightarrow S_2 = S_3$ if and only if S_3 is the relationship between PT_1 and PT_3 that can be deduced from the fact that S_1 is the relationship between PT_1 and PT_2 and S_2 the relationship between PT_2 and PT_3 (Figure 1).

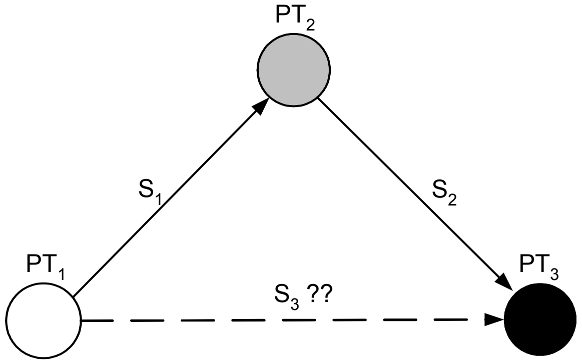


Figure 1: Concatenation of relationships

The results from this operator can be described in a 32 x 32 table.

Examples

$\{\equiv\} \rightarrow S = S$ for every S
$\{\equiv, \subset\} \rightarrow \{\subset\} = \{\subset\}$
$\{\supset\} \rightarrow \{\supset\} = \{\supset\}$ but
$\{\subset\} \rightarrow \{\oplus\} = \{\subset, \oplus, !\}$ and
$\{\oplus\} \rightarrow \{\oplus\} = \{\equiv, \subset, \supset, \oplus, !\}$.

The example data (box 1) assert that
$PT_1 \subset PT_4$ and $PT_1 \supset PT_6$.
Therefore
$PT_4 \supset PT_1$ and $PT_6 \subset PT_1$.
Now we can ask about the relationships between PT_4 and PT_8 on the one side and between PT_6 and PT_8 on the other side. Both of them follow from the already known relationships:
If PT_4 S_1 PT_1, PT_6 S_2 PT_1 and PT_1 S_3 PT_8 so are $S_1 = \{\supset\}$, $S_2 = \{\subset\}$ and $S_3 = \{\oplus\}$.
If the relationship which arises through concatenation between PT_4 and PT_8 is called S_4 (and S_5 the one between PT_6 and PT_8), so
$S_4 = S_1 \rightarrow S_3$ ($S_4 = \{\supset\} \rightarrow \{\oplus\}$)
which means

$S_4 = \{\supset, \oplus\}$

or, expressed differently, that

$PT_4 \{\supset, \oplus\} PT_8$

and

$S_5 = S_2 \rightarrow S_3$ ($S_5 = \{\subset\} \rightarrow \{\oplus\}$)

which means

$S_5 = \{\subset, \oplus, !\}$

or, in other words

$PT_6 \{\subset, \oplus, !\} PT_8$.

If we take into account that according to KOPERSKI & al. (2000) "*R. heterostichum* var. *gracilescens* is to be included in the synonymy of *R. sudeticum*", i.e. $PT_1 \oplus PT_8$, it is possible to say that both PT_6 and PT_8 include the type *R. heterostichum var. gracilescens* and therefore cannot exclude each other.

It can be therefore asserted that actually $PT_6 \{\subset , \oplus\} PT_8$ but that $PT_4 \{\supset, \oplus\} PT_8$ remains. It can also be deduced that $PT_5 \{\subset , \oplus, !\} PT_8$.

Since the "doubtful" marker is always transmitted to the derived relationships when "concatenated", the relationship between PT_4 and PT_8 is $S_4? = \{\supset, \oplus\}?$ and not merely $S_4 = \{\supset, \oplus\}$.

The potential taxon graph

We can conceive potential taxa and the relationships between the corresponding taxonomic concepts as an oriented graph (cf. BEACH & al., 1993), where the nodes are the potential taxa and where the set relationships assigned from expert(s) between two potential taxa build the oriented edges (Figure 2).

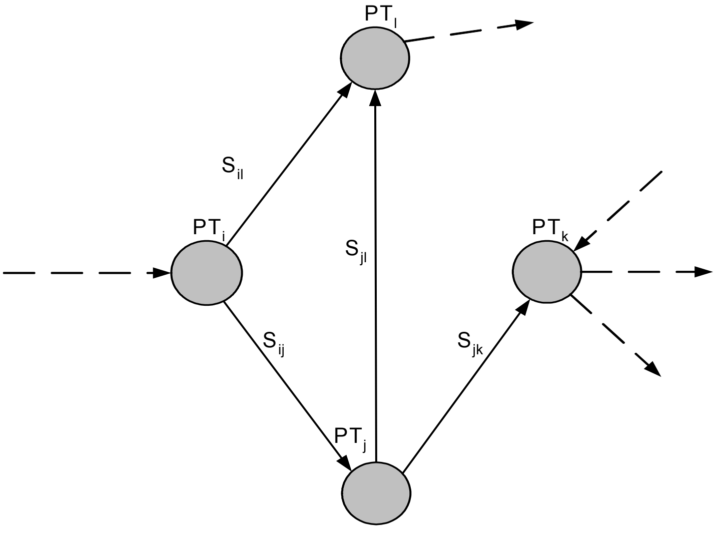


Figure 2: Nodes and edges in the potential taxon graph

The particularity of this potential taxon graph is that a relationship is assigned to every edge. As usual we define a path as a finite sequence of contiguous edges. Every path has an initial node and a terminal node.

Actually we have already discussed the case of assigning a relationship not to an edge but to a path consisting of two edges (the "concatenation" operator). But we have to generalise this matter by asking how to assign a combined relationship to a path consisting of any number of edges (that is for any path length). The answer relies on the iteration of the concatenation (Figure 3).

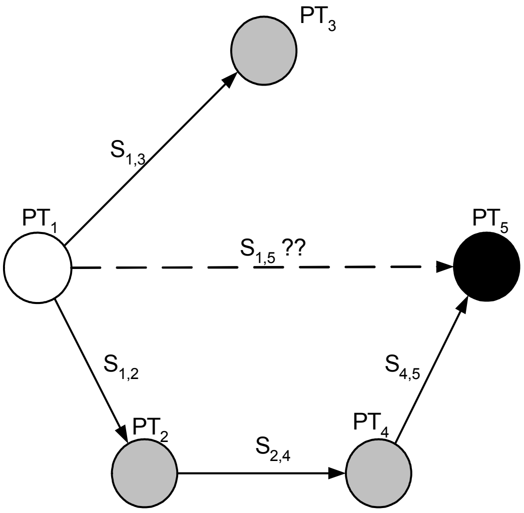


Figure 3: Concatenation in the potential taxon graph

Simultaneous and generalised paths

At first glance it seems now possible to automatically establish machine generated relationships between any two potential taxa, provided that there is a path in the potential taxon graph with the corresponding initial and terminal nodes. Gaps the experts left while establishing relationships could thus be closed. Unfortunately, there is no guaranty that there is only one path between two given nodes, on the contrary, as soon as some data accumulate in the system a variety of paths with the same initial and terminal nodes arise. We designate such a set of paths with the term “simultaneous paths”, while there should also be a “generalised path” that bundles all existing paths between two nodes (Figure 4). How to calculate the generalised path and the resulting combined relationship?

It seems reasonable to assign the combined relationship, which is the set intersection of all the combined relationships assigned to the paths belonging to the “generalised path” (remember that a combined relationship is a set of basic relationships). This essentially presumes that we equally trust all the experts who have contributed; we assume that the all combined relationships assigned to each of the paths must include the “real” relationship between the two potential taxa (nodes). Alternatively we could assume that the “real” relationship belongs to at least one of the combined relationships assigned to one of the simultaneous paths. This is the same as to rely on the opinions of all experts as a whole but not necessarily on the opinion of each of them. In this case we must use the set union (and not the set intersection) of all the combined relationships assigned to the paths for computing the relationship to be assigned to the “generalised path”.

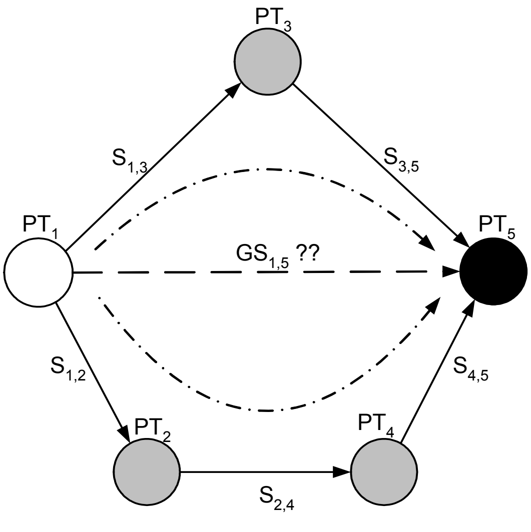


Figure 4: Generalised path

Moreover, the number of paths involved in “generalised” paths is bigger as it would normally be expected because of the fact that to each oriented relationship between the potential taxa PT$_1$ and PT$_2$ there exists implicitely another oriented relationship between PT$_2$ and PT$_1$, which is nothing else as the result of a reversal operator. An extract from a potential taxon graph thus looks like Figure 5.

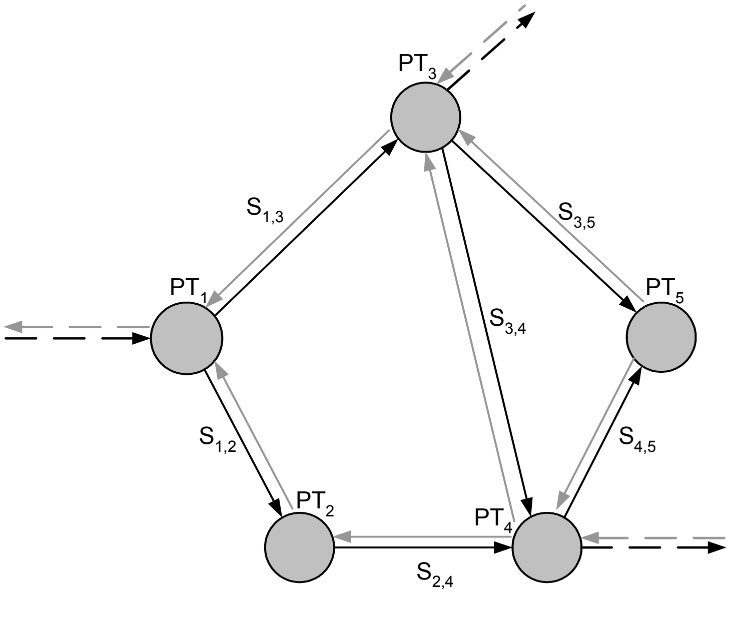


Figure 5: Extract from a potential taxon graph

Two further facts must be considered. First, cycles (paths beginning and ending in the same node) may occur in such a graph and must be excluded from the calculation. Second, there can be more than one edge between two nodes if different experts express different opinions about the relationship between the two underlying concepts. In this last case the discussion on “generalised paths” (see above) must be applied to what we could call “generalised edges”.

The mathematical-abstract representation

The formal description of the main items we handled so far is the following:

Let PT = $\{PT_1, ..., PT_n\}$ represent the potential taxa (i. e. sets of elements), which correspond to the taxonomic concepts.

Let S = $\{S_1, ..., S_{64}\}$ be 6-tuples, the components S_i of which are boolean (“true” = 1 and “false” = 0). The first five components represent the basic relationships R1 to R5 and the last one represents the “doubtful” flag. Any opinion about a relationship between two potential taxa can thus be described by means of one of the 64 different S_i.

Let $E \subset PT \times PT$ be the set of ordered pairs of potential taxa, to which experts assigned a (combined) relationship. Since to every oriented relationship there is an associated reverse relationship we have: $(PT_i, PT_j) \in E \Leftrightarrow (PT_j, PT_i) \in E$.

Let $f : E \rightarrow S$ be the function which describes the assignation of a combined relationship to an ordered pair of potential taxa.

As we already showed this amounts to a graph, where PT is the set of nodes, E the set of edges and S a set of values, which characterise each edge.

Let further $P_{i,t}$ be a path from an initial taxonomic concept PT_i to a terminal taxonomic concept PT_t , $GP_{i,t}$ be the generalised path with the same initial and terminal nodes and P be the set of all paths within the graph. $P_{i,t}$ is nothing else as an array of taxonomic concepts $(PT_1, \ldots, PT_n)$ where $PT_1 = PT_i$, $PT_n = PT_t$, $j \in (1,n\text{-}1) \Rightarrow (PT_j, PT_{j+1}) \in E$ and $j,k \in (1, ..., n) \Rightarrow ((PT_j = PT_k) \Rightarrow j = k)$.

The last two conditions correspond to the rule that only consecutive edges are allowed for building paths and that a path may traverse no node more than once.

Our aim is to find a general function $g : PT \times PT \rightarrow S$, which depends on f , on the architecture of the graph and on the set theory applied to the $S_i \in S$ when S_i is conceived as a set of basic relationships. As the result, g must provide the combined relationship existing between any two arbitrary taxonomic concepts in the graph.

Five steps are necessary to solve this problem:

1. Define the “concatenation” function $h : S \times S \rightarrow S$ through:
   $(f(PT_i, PT_j) = S_{ij} \wedge f(PT_j, PT_k) = S_{jk}) \Rightarrow h(S_{ij}, S_{jk}) = S_{ik}$, where S_{ik} is the combined relationship between PT_i and PT_k, which arises from the set theory and from the combined relationships S_{ij} (between PT_i and PT_j) and S_{jk} (between PT_j and PT_k).
   If $(PT_1, PT_2) \in E$ and $(PT_2, PT_3) \in E$, then the expression:
   $h(f(PT_1, PT_2), f(PT_2, PT_3))$ can be calculated.
2. Define an algorithm, which gives back the set $GP_{x,y}$ of all paths $P_{x,y}$ between two given PT_x and $PT_{y.}$
3. Define the generalised concatenation function $h' : P \rightarrow S$, which assigns a relationship to a path, through:
   $((PT_{1,2}) = (PT_1, PT_2) \in E) \Rightarrow h'(P) = f(PT_1, PT_2)$ and
   $((PT_{1,n}) = (PT_1, ..., PT_n) \in P) \Rightarrow h'(P) = h(.....(h(f(PT_1, PT_2), f(PT_2, PT_3)),.....), f(PT_{n\text{-}1}, PT_n))$

4. Define a function $h^* : S \times S \rightarrow S$ through:

$h^* (S_i, S_j) = S_k \Leftrightarrow (((R_{i6} = 0 \wedge R_{j6} = 1) \Rightarrow (S_k = S_i)) \wedge$

$((R_{i6} = 1 \wedge R_{j6} = 0) \Rightarrow (S_k = S_j)) \wedge$

$((R_{i6} = R_{j6}) \Rightarrow ((R_{k6} = R_{i6}) \wedge n \in \{1,...,5\} \Rightarrow R_{kn} = (R_{in} \wedge R_{jn})))),$

where $S_i = (R_{i1},....,R_{i6})$, $S_j = (R_{j1},....,R_{j6})$ and $S_k = (R_{k1},....,R_{k6})$.

This function models the set intersection operator we discussed in case of simultaneous paths.

Define another function $h^{**} : S \times S \rightarrow S$ through:

$h^{**} (S_i, S_j) = S_k \Leftrightarrow (((R_{i6} \neq R_{j6}) \Rightarrow (R_{k6} = 1)) \wedge ((R_{i6} = R_{j6}) \Rightarrow (R_{k6} = R_{i6})) \wedge$

$n \in \{1,...,5\} \Rightarrow R_{kn} = (R_{in} \vee R_{jn})))).$

This function models the set union operator we discussed in case of simultaneous paths.

5. Define the function $g : PT \times PT \rightarrow S$

through:

$g (PT_x, PT_y) = h^* (....(h^* (h' (P_{x,y,1}), h' (P_{x,y,2})),...), h' (P_{x,y,n}))$

where $GP_{x,y} = \{P_{x,y,i}\}$ and $i \in \{1,...,n\}$

if “set intersection” is chosen for handling “simultaneous paths”

or through:

$g (PT_x, PT_y) = h^{**} (....(h^{**} (h' (P_{x,y,1}), h' (P_{x,y,2})),...), h' (P_{x,y,n}))$

where $GP_{x,y} = \{P_{x,y,i}\}$ and $i \in \{1,...,n\}$

if “set union” is chosen.

For the consistency of the evaluations the following conditions must be (and are) fulfilled:

Commutativity:

$h^* (S_i, S_j) = h^* (S_j, S_i)$

$h^{**} (S_i, S_j) = h^{**} (S_j, S_i)$

Associativity:

$h (h (S_i, S_j), S_k)) = h (S_i, h (S_j, S_k))$

$h^* (h^* (S_i, S_j), S_k)) = h^* (S_i, h^* (S_j, S_k))$

$h^{**} (h^{**} (S_i, S_j), S_k)) = h^{**} (S_i, h^{**} (S_j, S_k))$

Distributivity:

$h (h^* (S_i, S_j), S_k) = h^* (h (S_i, {}_{S}k), h (S_j, S_k))$

$h (h^{**} (S_i, S_j), S_k) = h^{**} (h (S_i, S_k), h (S_j, S_k))$

$h (S_i, h^* (S_j, S_k)) = h^* (h (S_i, S_j), h (S_i, S_k))$

$h (S_i, h^{**} (S_j, S_k)) = h^{**} (h (S_{i,} S_j), h (S_i, S_k))$

For completion and simplification we define three additional functions:

The reversal function for basic relationships $f' : R \rightarrow R$ through:

$f' (Ri) = Ri$ for i= 1,4 or 5 and $f' (R2) = R3$ and $f' (R3) = R2$

The reversal function for combined relationships $f'' : S \rightarrow S$ through:

$f'' (S_1) = S_2 \Leftrightarrow S_2 = ((f' (a_1), \dots f' (a_5), a_6),$

where $S_1 = (a_1,...., a_5,a_6)$

The negation function $f''' : S \rightarrow S$ through:

$f''' (S_1) = S_2 \Leftrightarrow S_2 = (\neg a_1, \dots ,\neg a_5, a_6),$

where $S_1 = (a1,...., a5,a6)$.

The operator rules for relationships

For the formal description of rules we use Visual Basic as an example to define a "relationship data type" and to describe the operator rules for relationships, that is the reversal rule f’’, the negation rule f’’’, the intersection rule h*, the union rule h** and the concatenation rule h.

Definition of a datatype for "combined relationship"-objects:

```
Public Type Relationship
  Congruent_to As Boolean
  Is_included_in As Boolean
  Includes As Boolean
  Overlaps As Boolean
  Excludes As Boolean
  Doubtful As Boolean
End Type
```

Reversal rule for "combined relationships":

```
Public Function reverse(Rel1 As Relationship) As Relationship
  reverse = Rel1
  reverse.Is_included_in = Rel1.Includes
  reverse.Includes = Rel1.Is_included_in
End Function
```

Negation rule for "combined relationships":

```
Public Function negation(Rel1 As Relationship) As Relationship
  negation.Congruent_to = Not Rel1.Congruent_to
  negation.Is_included_in = Not Rel1.Is_included_in
  negation.Includes = Not Rel1.Includes
  negation.Overlaps = Not Rel1.Overlaps
  negation.Excludes = Not Rel1.Excludes
  negation.Doubtful = Rel1.Doubtful
End Function
```

Unification rule for two "combined relationships" (strong agreement - intersection):

```
Public Function cons(Rel1 As Relationship, Rel2 As Relationship) As Relationship
  If Rel1.Doubtful = Rel2.Doubtful Then
    cons.Congruent_to = Rel1.Congruent_to And Rel2.Congruent_to
    cons.Is_included_in = Rel1.Is_included_in And Rel2.Is_included_in
    cons.Includes = Rel1.Includes And Rel2.Includes
    cons.Overlaps = Rel1.Overlaps And Rel2.Overlaps
    cons.Excludes = Rel1.Excludes And Rel2.Excludes
    cons.Doubtful = Rel1.Doubtful
  ElseIf Rel1.Doubtful = False Then
    cons.Congruent_to = Rel1.Congruent_to
    cons.Is_included_in = Rel1.Is_included_in
    cons.Includes = Rel1.Includes
    cons.Overlaps = Rel1.Overlaps
    cons.Excludes = Rel1.Excludes
    cons.Doubtful = Rel1.Doubtful
  Else
    cons.Congruent_to = Rel2.Congruent_to
    cons.Is_included_in = Rel2.Is_included_in
    cons.Includes = Rel2.Includes
    cons.Overlaps = Rel2.Overlaps
    cons.Excludes = Rel2.Excludes
    cons.Doubtful = Rel2.Doubtful
  End If
End Function
```

Unification rule for two "combined relationships" (weak agreement - union):

```
Public Function large_cons(Rel1 As Relationship, Rel2 As Relationship) As Relationship
  large_cons.Congruent_to = Rel1.Congruent_to Or Rel2.Congruent_to
  large_cons.Is_included_in = Rel1.Is_included_in Or Rel2.Is_included_in
  large_cons.Includes = Rel1.Includes Or Rel2.Includes
  large_cons.Overlaps = Rel1.Overlaps Or Rel2.Overlaps
  large_cons.Excludes = Rel1.Excludes Or Rel2.Excludes
  large_cons.Doubtful = Rel1.Doubtful Or Rel2.Doubtful
End Function
```

Concatenation rule for two contiguous "combined relationships":

```
Public Function concatenate(Rel1 As Relationship, Rel2 As Relationship) As Relationship
Dim RelNull As Relationship
Dim RelFull As Relationship
Dim TempRelResult As Relationship
  RelNull.Congruent_to = False
  RelNull.Is_included_in = False
  RelNull.Includes = False
  RelNull.Overlaps = False
  RelNull.Excludes = False
  RelNull.Doubtful = False
  RelFull.Congruent_to = True
  RelFull.Is_included_in = True
  RelFull.Includes = True
  RelFull.Overlaps = True
  RelFull.Excludes = True
  RelFull.Doubtful = False
  concatenate = RelNull
  TempRelResult = RelNull
  If Rel1.Congruent_to Then
    concatenate = Rel2
  End If
  If Rel2.Congruent_to Then
    TempRelResult = Rel1
    concatenate = large_cons(concatenate, TempRelResult)
    TempRelResult = RelNull
  End If
  If Rel1.Is_included_in Then
    If Rel2.Is_included_in Then
      TempRelResult.Is_included_in = True
      concatenate = large_cons(concatenate, TempRelResult)
      TempRelResult = RelNull
    End If
    If Rel2.Includes Then
      TempRelResult = RelFull
      concatenate = large_cons(concatenate, TempRelResult)
      TempRelResult = RelNull
    End If
    If Rel2.Overlaps Then
      TempRelResult.Is_included_in = True
      TempRelResult.Overlaps = True
      TempRelResult.Excludes = True
      concatenate = large_cons(concatenate, TempRelResult)
      TempRelResult = RelNull
    End If
    If Rel2.Excludes Then
      TempRelResult.Excludes = True
      concatenate = large_cons(concatenate, TempRelResult)
      TempRelResult = RelNull
    End If
  End If
  If Rel1.Includes Then
  If Rel2.Is_included_in Then
    TempRelResult.Congruent_to = True
    TempRelResult.Is_included_in = True
    TempRelResult.Includes = True
    TempRelResult.Overlaps = True
    concatenate = large_cons(concatenate, TempRelResult)
    TempRelResult = RelNull
  End If
  If Rel2.Includes Then
    TempRelResult.Includes = True
    concatenate = large_cons(concatenate, TempRelResult)
    TempRelResult = RelNull
  End If
  If Rel2.Overlaps Then
    TempRelResult.Includes = True
    TempRelResult.Overlaps = True
    concatenate = large_cons(concatenate, TempRelResult)
    TempRelResult = RelNull
  End If
  If Rel2.Excludes Then
    TempRelResult.Includes = True
    TempRelResult.Overlaps = True
    TempRelResult.Excludes = True
    concatenate = large_cons(concatenate, TempRelResult)
    TempRelResult = RelNull
  End If
End If
If Rel1.Overlaps Then
  If Rel2.Is_included_in Then
    TempRelResult.Is_included_in = True
    TempRelResult.Overlaps = True
    concatenate = large_cons(concatenate, TempRelResult)
    TempRelResult = RelNull
  End If
  If Rel2.Includes Then
    TempRelResult.Includes = True
    TempRelResult.Overlaps = True
    TempRelResult.Excludes = True
    concatenate = large_cons(concatenate, TempRelResult)
    TempRelResult = RelNull
  End If
  If Rel2.Overlaps Then
    TempRelResult = RelFull
    concatenate = large_cons(concatenate, TempRelResult)
    TempRelResult = RelNull
  End If
  If Rel2.Excludes Then
    TempRelResult.Includes = True
    TempRelResult.Overlaps = True
    TempRelResult.Excludes = True
    concatenate = large_cons(concatenate, TempRelResult)
    TempRelResult = RelNull
  End If
End If
If Rel1.Excludes Then
  If Rel2.Is_included_in Then
    TempRelResult.Is_included_in = True
    TempRelResult.Overlaps = True
    TempRelResult.Excludes = True
    concatenate = large_cons(concatenate, TempRelResult)
    TempRelResult = RelNull
  End If
  If Rel2.Includes Then
```

```
        TempRelResult.Excludes = True
        concatenate = large_cons(concatenate, TempRelResult)
        TempRelResult = RelNull
      End If
      If Rel2.Overlaps Then
        TempRelResult.Is_included_in = True
        TempRelResult.Overlaps = True
        TempRelResult.Excludes = True
        concatenate = large_cons(concatenate, TempRelResult)
        TempRelResult = RelNull
      End If
      If Rel2.Excludes Then
        TempRelResult = RelFull
        concatenate = large_cons(concatenate, TempRelResult)
        TempRelResult = RelNull
      End If
    End If
    concatenate.Doubtful = Rel1.Doubtful Or Rel2.Doubtful
  End Function
```

The functions here documented have been included in an experimental software tool, which can be downloaded from the project pages (GEOFFROY, 2001).

References cited

BEACH, J. H., PRAMANIK, S. & BEAMAN, J. H. (1993): Hierarchic taxonomic databases. Ch. 15 (pp. 241-256) in: FORTUNER, R. (ed.): Advances in computer methods for systematic biology: artificial intelligence, databases, computer vision. John Hopkins University Press, Baltimore.

CORLEY, M. F. V, CRUNDWELL, A. C., DÜLL, R., HILL, M. O. & SMITH, A. J. E. (1981): Mosses of Europe and the Azores; an annotated list of species, with synonyms from the recent literature. J. Bryol. 11(4): 609-689.

CORLEY, M. F. V, CRUNDWELL, A. C. (1991): Additions and amendments of the mosses of Europe and the Azores. J. Bryol. 16(3): 337-356.

FRAHM, J.-P. & FREY, W. (1992): Moosflora. 3. Aufl. – Stuttgart (Ulmer) – Uni-Taschenb. 1250, 528 pp.

Geoffroy, M. (2001 [Jan 15 2003]): MoReTax Protoytpe http://www.bgbm.org/BioDivInf/ Projects/ MoreTax.

GEOFFROY, M. & BERENDSOHN, W. G. (2003): The concept problem in taxonomy: importance, components, approaches. Schriftenreihe Vegetationsk. 39: 5-14.

KOPERSKI, M., SAUER, M., BRAUN, W. & GRADSTEIN, S. R. (2000): Referenzliste der Moose Deutschlands. Schriftenreihe Vegetationsk. 34: 1-519.

LUDWIG, G., DÜLL, R., PHILIPPI, G., AHRENS, M., CASPARI, S. KOPERSKI, M., LÜTT, S, SCHULZ, F. & SCHWAB, G. (1996): Rote Liste der Moose (Anthocerophyta et Bryophyta) Deutschlands. In: LUDWIG, G. & SCHNITTLER, M. [Bearb.]: Rote Liste der gefährdeten Pflanzen Deutschlands. – Hiltrup (Landwirtschaftsverl.) – Schriftenreihe Vegetationsk. 28: 189-306.

MÖNKEMEYER, W. (1927): Die Laubmoose Europas. Andreales – Bryales. In: RABENHORST, G. L. [Begr.]: Kryptogamenflora von Deutschland, Österreich und der Schweiz. Bd. IV. – Leipzig (Geest & Portig) 960 pp.

SMITH, A. J. E. (1980): The mossflora of Britain and Ireland. – Cambridge, Mass. (Cambridge University Pr.) 706 pp.

Transmission of taxon-related factual information

MARC GEOFFROY & WALTER G. BERENDSOHN [b]

Preceding articles in this volume have dealt with the necessity of a transmission engine to overcome the uncertainties of names as an index to biological information (GEOFFROY & BERENDSOHN, 2003), with the design of the core concept-based taxonomic database (BERENDSOHN & al., 2003), its data entry and editing functionality (GÜNTSCH & al., 2003), and its extension to fully cover botanical nomenclature (KUSBER & al., 2003), as well as with the intricacies of concept relations within the potential taxon graph (GEOFFROY & GÜNTSCH, 2003). The "factual information" linked to taxon names covers:

- Uses (mostly human) of the organism or parts of the organism and threats (to species itself, to hosts, to health, to environment, etc.)
- Ecology (pollination, symbiosis, parasitism, indicator value, edaphic and climatic requirements, etc.) of the organism
- Geographical range or occurrence of the organism
- Molecular data (natural substances, genes, sequences, physiology, etc.) derived
- Other descriptive data

Apart from user and provider requirements, the two main factors influencing the processing and transmission of factual information are the involved concept relationships and certain properties of the factual information itself (the "applicability").

The influence of concept relationships

Users who wish to get as many facts as possible concerning a taxon need a search engine that retrieves all facts from all relevant accessible sources. The user query as well as the factual information is linked to taxonyms, and as shown it is possible to deduce the relationship between two potential taxa (represented by taxonyms) as long as these are connected through a path within the potential taxon graph. Relevant sources are those that provide information linked to a node that has a relationship with the queried node that is meaningful in the context of the query.

Retrieving factual information would therefore be the systematic gathering of all facts linked to all potential taxa that are connected through paths. It is possible to envisage this as information travelling over paths from one node of the potential taxon graph to another, allowing the concentration of all facts in any one of those nodes.

However, there are a number of caveats, which have to be observed and analysed. The accuracy of facts depends on the relationships involved in the transmission. If, for example, the concept of the taxon on the querying side is wider than that used in the factual database, the user must be alerted that the factual information retrieved may not apply to all elements of the potential taxon queried. If users are not alerted about this situation, they will assume that all gathered facts could be directly applied to the taxon they used to launch the query and thus be freely combined. This clearly may lead to false conclusions. On the other hand it would be frustrating for the user getting the complete amount of factual information and at the same time the warning that he'd better not use it further since it cannot be trusted. Therefore the engine must give back not only the retrieved facts but also some kind of measure of accuracy for each of the facts.

Suppose that some source A asserts that the species "X" is poisonous and that an expert asserts that there exists a relationship $R_{xa,xb}$ between the potential taxon X sec. A, which is associated to the taxon name "X" in the source A, and the potential taxon X sec. B, which is associated to the taxon name "X" in source B. Is it possible to deduce that the fact "is poisonous" applies also to X sec. B? The answer depends on the relationship $R_{xa,xb}$:

- If this relationship is R1 ("congruent") or R3 ("includes") then it must be concluded that "X sec. B" is also poisonous.
- If this relationship is R5 ("excludes") then nothing can be deduced for "X sec. B".
- If this relationship is R2 ("included in") or R4 ("overlaps") then it is only possible to deduce that some elements in "X sec. B" are poisonous.

It is thus not enough to consider the alternative whether a given fact applies or not to a concept. The "degree" in which factual information applies to a concept must be qualified. Moreover, depending on the user group, certain information may be filtered out, and the verbal expression of caveats for the user interface may vary.

Different categories for the applicability of factual information

The second major factor influencing the transmission of factual information from its source through the transmission engine to the user is what we call "applicability". In essence this is the relation between properties of the set and properties of the elements in the set, i.e. the relationship between information connected to the entire potential taxon (the set) and the individual elements within that potential taxon. We distinguish five categories of information applicability. The information is:

- Fully applicable if it applies to every element of the potential taxon
  E.g. a source states that species A has blue petals, this information is fully applicable .
- Partially applicable if it applies to some elements (a subset) of the potential taxon
  E.g. a source states that plants of the species B were found to contain commercially interesting levels of a certain natural substance only in some populations. However, note that we delimit the concept of partial applicability by means of the certainty that the imformation applies to some elements, but we do not exclude the possibility that it applies to all.
- Doubtfully applicable if it may apply to some elements of the potential taxon
  E.g. a source states that there have been unconfirmed reports of poisoning from ingestion of the fruits of species C.
- Not applicable if there is absolutely no reason why the information should apply to any element of the potential taxon
  Explicitly negated infomation belongs here. This could also be used to introduce negation of fully applicable information (e.g. the conclusion that – if flowers have 5 petals – they do not have 3, 4, or 6, etc.).
- Set applicable if it constitutes a summary of information about the individuals which cannot be directly applied to individual elements.
  Most taxon-level descriptive information belongs here, same as information related to geographical distribution (perhaps the clearest example: the information that species D has a distributional range from the Iberian Peninsula to Russia cannot be applied to the individual organism).

The transmission engine computes the applicability of information provided by a source node (a taxonym in a factual database) to a target node (the queried potential taxon), based on the relationship given in the potential taxon graph and on the original applicability category of the information. The example of species X given above illustrates a simple case where the transmission process actually may change the applicability.

The fact "is poisonous" is fully applicable to X sec. A. Its applicability to X sec. B depends on the relationship $R_{1,2}$:

- If this relationship is R1 ("congruent) or R3 ("includes") then it is fully applicable.
- If this relationship is R5 ("excludes") then it is not applicable.
- If this relationship is R2 ("included in") or R4 ("overlaps") then it is partially applicable.

Some general rules can be formulated for the transmission of applicablity. The simplest case is if the relationship between PT_1, PT_2, and PT_3 was defined as congruent, because then all information of all applicability categories can be directly transmitted between them, and the transmission process does not influence the applicability.

It should be noted, however, that the engine as presently devised is blind with respect to the semantics and structure of the information. For example, the engine is unable to identify logical contradictions such as when facts F_1 and F_2 are fully applicable to a potential taxon PT but F_1 asserts that "flowers are blue" and F_2 asserts, "flowers are white".

Neither can the engine draw conclusions and produce new facts by interpreting the orginal ones. For example, the information on potential taxon Y that it "occurs in Spain and Portugal" is only set applicable. However, it can be interpreted positively as "there are elements of X occuring in Spain and others occurring in Portugal" (creating partially applicable data) or negatively as "there is no element occuring outside the Iberian Peninsula" (not applicable). Similar problems are encountered in the generalisation of descriptive information (the petal length of an individual is not the range given for the species, but the individual should not have a petal length outside the range given, etc.). These are simple examples, but the ongoing discussion of a standard data format for descriptive biological information has shown the complexity of the terminology and structures involved (see HAGEDORN, 2002). Interpretation of biological facts would have to be based on standard structures and could be a task for a semantic network (see e.g. HEFLIN, 2001), but this is clearly out of scope for the project here presented. However, we see no problem in interfacing with such a semantic network, actually treating it as a factual database, or using it as part of the interface with factual databases.

On the other hand, set applicability poses a problem in the transmission process as soon as no clear-cut congruency or exclusion exists between the initial and terminal potential taxa. Clearly, we would be better off if information would always be connected to individual elements (e.g. specimens), because then a much simpler definition of a taxon concept can be used (the Prometheus Model, PULLAN & al. 2000). A specimen-based information system for descriptive taxonomic information is imaginable, though only in a somewhat distant future. However, for the time being and for much of the historical and extra-taxonomical information we will have to rely on (and work with) partial or set applicable information.

The applicability rules for the transmission of factual information

The applicability of transmitted factual information for the elements of a terminal potential taxon depends on the applicability of factual information to the elements of the initial potential taxon to which it was originally linked and on the combined relationship assigned to the generalised path between the initial and the terminal potential taxon.

Call PT_i the initial potential taxon, PT_t the terminal potential taxon and S the combined relationship assigned to the generalised path between both. Moreover call A_i the applicability category of some factual information linked to PT_i and A_t the corresponding category when transmitted to PT_t. The applicability rules for the transmission of factual information can be then formulated as:

- If A_i is 'fully applicable' and if every element of PT_t belongs also to PT_i then A_t is also 'fully applicable'.
  $((A_i = \text{'fully applicable'}) \wedge (x \in PT_t \Rightarrow x \in PT_i)) \Rightarrow A_t = \text{'fully applicable'}$
- If A_i is 'fully applicable' and if there is at least one common element to PT_i and PT_t and if at least one element of PT_t does not belong to PT_i then A_t is 'partially applicable'.
  $(A_i = \text{'fully applicable'} \wedge (R2 \in S \vee R4 \in S) \wedge R5 \notin S) \Rightarrow A_t = \text{'partially applicable'}$
- If A_i is 'partially applicable' and if every element of PT_i belongs also to PT_t then A_t is also 'partially applicable'.
  $(A_i = \text{'partially applicable'} \wedge (S = \{R1\} \vee S = \{R2\} \vee S = \{R1, R2\})) \Rightarrow A_t = A_i$
- If A_i is not 'not applicable' and if PT_i and PT_t may have no common element then A_t is 'doubtfully applicable'.
  $(A_i \neq \text{'not applicable'} \wedge R5 \in S \wedge S \neq \{R5\}) \Rightarrow A_t = \text{'doubtfully applicable'}$

Note that the category ‘doubtfully applicable’ makes only sense because we are allowing combined relationships (in particular combined relationships for which the basic relationship “exclusion” is only one of several possible basic relationships).

- If A_i is ‘set applicable’ and PT_i is congruent to PT_t then At is ‘set applicable’. (A_i = ‘set applicable’ $\wedge$ S = {R1}) $\Rightarrow$ At = ‘not applicable’)
- If A_i is ‘set applicable’ and if PT_i and PT_t are not congruent but may have a common element then A_t is ‘doubtfully applicable’. (A_i = ‘set applicable’ $\wedge$ S $\neq$ {R1} $\wedge$ S $\neq$ {R5}) $\Rightarrow$ A_t = ‘doubtfully applicable’)
- If A_i is ‘not applicable’ or PT_i and PT_t have no common element then A_t is ‘not applicable’. (A_i = ‘not applicable’ $\vee$ S = {R5}) $\Rightarrow$ A_t = ‘not applicable’)

How does path length influence the transmission of factual information and the modification of the applicability? Preliminary answers can be derived from the data of the “Checklist of German mosses” (KOPERSKI & al., 2000). About 75% of the concept relationships cited there are ‘congruent‘, 10% are ‘is included in‘, 10% are ‘includes‘, 5% are ‘overlaps‘, while the occurrence of the ‘excludes‘ relationships is insignificant. We assume (a) this distribution of basic relationships and (b) factual information that is always ‘fully applicable’ at the initial potential taxon. Applying the rules, we obtain the following probabilities for applicability of the transmitted factual information to a terminal potential taxon:

- 85% for ‘fully applicable’ and 15% for ‘partially applicable’ for paths with length 1 (actually edges)
- about 45% for ‘fully applicable’, about 40% for ‘partially applicable’ and about 15% for ‘doubtfully applicable’ for paths with length 5
- about 20% for ‘fully applicable’, about 35% for ‘partially applicable’ and about 45% for ‘doubtfully applicable’ for paths with length 10
- and still about 5% for ‘fully applicable’, about 15% for ‘partially applicable’ and about 80% for ‘doubtfully applicable’ for paths with length 20

Our conclusion is that the transmission engine can certainly help to handle factual information - provided that we have information about the concepts the facts are connected to.

References cited

BERENDSOHN, W. G., DÖRING, M., GEOFFROY, M., GLÜCK, K., GÜNTSCH, A., HAHN, A., KUSBER, W.-H., LI, J.-L., RÖPERT, D. & SPECHT, F. (2003): The Berlin Taxonomic Information Model. Schriftenreihe Vegetationsk. 39: 15-42.

GEOFFROY, M. & BERENDSOHN, W. G. (2003): The concept problem in taxonomy: importance, components, approaches. Schriftenreihe Vegetationsk. 39: 5-14.

GEOFFROY, M. & GÜNTSCH, A. (2003): Assembling and navigating the potential taxon graph. Schriftenreihe Vegetationsk. 39: 71-82.

GÜNTSCH, A., GEOFFROY, M., DÖRING, M., GLÜCK, K., LI, J.-J., RÖPERT, D., SPECHT, F. & BERENDSOHN, W. G. (2003): The taxonomic editor. Schriftenreihe Vegetationsk. 39: 43-56.

HAGEDORN, G. (2002 [Dec 30]) (convenor): TDWG working group, Structure of Descriptive Data.. http://www.tdwg.org/sddhome.html.

HEFLIN, J. (2001): Towards the Semantic Web: Knowledge Representation in a Dynamic, Distributed Environment. Ph.D. Thesis, University of Maryland, College Park. 250 pp.

KOPERSKI, M., SAUER, M., BRAUN, W. & GRADSTEIN, S. R. (2000): Referenzliste der Moose Deutschlands. Schriftenreihe Vegetationsk. 34: 1-519.

KUSBER, W.-H., GLÜCK, K., GEOFFROY, M. & JAHN, R. (2003): Typification – an extension of the Berlin Taxonomic Information Model. Schriftenreihe Vegetationsk. 39: 57-70.

PULLAN, M. R., WATSON, M. F., KENNEDY, J. B., RAGUENAUD, C & HYAM, R. (2000): The Prometheus Taxonomic Model: a practical approach to representing multiple classifications. Taxon 49:55-75.

Towards the implementation of the “transmission engine”

MARC GEOFFROY

The MoReTax project had the aim to summarise existing knowledge and to provide the theoretical groundwork for a software solution of the concrete difficulties users of taxonomic information are facing because of the concept problem in taxonomy. As shown for the taxonomic editor component (GÜNTSCH & al., 2003), successful development of a sound theoretical base is often impossible without some practical application. So, some of the features here introduced are in fact prototypic implementations. Nevertheless, these are intended to demonstrate possible solutions; they do not represent any final product.

Apart from the underlying information model for the taxonomic core database (BERENDSOHN & al., 2003), the design of the taxonomic editor (GÜNTSCH & al., 2003), and the rules for extracting concept relationships from traditional taxonomic treatments (BERENDSOHN, 2003), the following specifications have to be created for the implementation of the transmission engine:

1) Tuning the rules that are used by the transmission engine to calculate relationships
2) Tuning the output, i.e. the way the resulting information is presented to users
3) Defining the database (the extension to the Berlin Model) needed to store the configuration parameters defined
4) Designing the user interface for the “Rule Tuner”
5) Defining the database interfaces between the transmission engine and the factual databases
6) Designing an end-user interface to the transmission engine

1. Rule tuning

The core rules we need for the transmission process are the following:

- Operators on one combined relationship. These are the “reversal” and the “negation” operators.
- Operators on two combined relationships. These are the “concatenation”, the “intersection” and the “union” operators.
- The rule for building paths on the basis of edges.
- The rules for assigning combined relationships to paths and to generalised paths (including generalised edges).

  For the definition of these rules and functions see GEOFFROY & GÜNTSCH (2003).
- The applicability rules for the transmission of factual information.

  See GEOFFROY & BERENDSOHN (2003b) for these rules.
- A general break-off condition to avoid unrestricted navigation or processing of those paths in the potential taxon graph that actually do not contribute to clarify the relationship between two potential taxa. This is the case when the assigned combined relationship is R (the set of all basic relationships), because this means that nothing can be said about the real basic relationship and therefore about the applicability of information transmitted over this path.

An important aspect about concept relationships is that they are in part implicit in traditional taxonomical relations (see BERENDSOHN, 2003). In this context the relationships implied in classification trees are very important. For each lower ranking taxon an 'included in' relationship with the corresponding higher taxon can be automatically generated if the source of the two is the same. The same holds true for the reverse relationship 'includes'. As a result, for each classified potential taxon in the graph, we would get paths beginning at the lowest rank treated and ending at the highest rank. The "general break-off condition" takes care of the fact that such a path cannot be followed "up and down" the tree because this leads to the concatenation of a 'included in' with an 'includes' relation, which results in the set R (all possible relationships).

For practical purposes, the application of these rules has to be further adjusted or limited. Conditions may be set and applied that influence the actual output of the engine for different user groups and which allow fine-tuning this output according to user demands. These conditions amount to a set of parameters, which affect the concrete functioning of the "transmission engine".

The "generalised edge" issue

We have already seen that different experts (or sources) could have different opinions about the relationship between the same pair of potential taxa. If these opinions are considered as being equivalent then assigning a single relationship to the "generalised edge" can be done either by using the union rule or by using the intersection rule on all combined relationships describing the relationship between the two potential taxa. (This follows the same procedure as assigning a single relationship to a "generalised path".) So the first choice that has to be made concerns the standard rule that should be applied as a default for this process. This parameter is called the **StandardSetOperator**.

In the following, attributes for parameters used by the Rule Tuner are written in boldface where introduced. For other conventions see the section on "Methods and conventions" in Berendsohn & al. (2003).

If for some reason some kind of preference hierarchy for the sources of relationships is introduced, we need further adjustments. Such a preference hierarchy can be described by giving each expert (or better: each relationship source) a **Weight**. These weights may even depend on the taxonomic group the concerned potential taxa belong to, since experts are often specialised in certain taxonomic groups.

Explicitly weighting expert opinion is of course a somewhat delicate issue, which, however, cannot be circumvented in a system as proposed here. It should be considered that this has always been part of the research process (the "authoritative treatments", etc.) and it has always been up to the actual integrator (or editor) of the information to weigh the opinion of their peers. In the case of the transmission engine, this will be done by taxonomic system managers, who will base their decision on their own knowledge and on the specific requirements of their system.

Weights are defined as numbers ranging from 0 to 1 where 1 means highest preference (i.e. always take this opinion into account). The weight assigned to a source for a taxonomic group is assigned, as a new attribute, to any edge (concerning potential taxa within this taxonomic group) the source has created. If, for instance, a source E is weighted with 0.8 for the taxonomic group G and if it asserts that the potential taxon PT_1 'is included in' the potential taxon PT_2 (both potential taxa belonging to G), then the corresponding edge (PT_1,PT_2) will not only be characterised by the source E and the relationship R2 but also by the weight 0.8. The default value for Weight is 0 so that a sources opinion is only taken account of if it has been explicitly designated for the purpose in a configuration setting (see below).

Weighting by taxonomic group presumes, however, that some standard reference taxonomy exists by which these groups can be assigned. The degree of cover needed varies between organism groups but normally taxa from the generic level upwards would be the maximum cover necessary.

With this new attribute for edges it is possible to fine-tune the assignation of relationships to "generalised edges". First of all it is possible to establish a minimum weight (the **ExcludeWeight**) below which edges are not taken into account for the calculation of the generalised edge. Suppose, for instance, that three sources E_1, E_2 and E_3 have different opinions about the relationship between PT_1 and PT_2: E_1 asserts that the relationship is $\{\equiv, \subset\}$, E_2 asserts that the relationship is $\{\equiv\}$ and E_3 asserts that the relationship is $\{\equiv, \supset\}$. Suppose furthermore, that the weights assigned to sources are 0.9 for E_1, 0.7 for E_2 and 0.5 for E_3. If we ignore weights we would deduce that the resulting relationship for the generalised edge (PT_1,PT_2) is $\{\equiv, \subset, \supset\}$ in case of 'union' and $\{\equiv\}$ in case of 'intersection'. If we decide to take 0.75 as the value for ExcludeWeight then the only remaining edge for our calculation is the one associated with the source E_1, which means that the assigned relationship to (PT_1,PT_2) remains $\{\equiv, \subset\}$ in any case.

But if we take 0.6 for ExcludeWeight then the resulting relationship for the generalised edge (PT_1,PT_2) is $\{\equiv, \subset\}$ in case of 'union' and $\{\equiv\}$ in case of 'intersection'.

An additional feature is to differentiate weights depending on the distribution of basic relationships a certain source assigns, so that differing tendencies of sources in the process of comparing concepts could be levelled out. Although this was considered to be a marginal issue for the time being, attributes for these parameters have been included in the model (**WeightForCongruency,** etc.).

It is also possible to restrict the number of edges to be involved in the assignment process by establishing a maximum **StandardDistance** for the weights from the value of the maximum weight encountered among the edges.

Suppose that in our example the ExcludeWeight is fixed to 0.2 and that the StandardDistance is established to, say, 0.3, then only the edges with the weights 0.9 and 0.7 will be used for the assignation of a relationship to the "generalised edge", because the difference between 0.7 and the maximum weight (0.9) is smaller than 0.3. Had the StandardDistance been established as 0.1, only the edge with the Weight 0.9 would have been involved in the calculation, since the difference between 0.7 and the maximum weight (0.9) is greater than 0.1. If the StandardDistance had been 0.5, then all three edges would have been involved, since the difference between 0.5 and the maximum weight (0.9) is smaller than 0.5.

The higher the maximum weight of an edge the more reliable is the corresponding relationship and therefore the less is the need to take in account relationships of other edges with lower weights in the calculation. Therefore it could be useful to choose the distance and/or the set operator to be applied (union or intersection) depending on the weights of the concerned edges. More precisely, it can make sense to establish that if the maximum weight of all concerned edges lies within a given range of values (the **IntervalForMaximumWeight**), a certain **Distance** and/or a certain **SetOperator** will be automatically chosen. It should thus be possible to define several **IntervalsForMaximumWeight,** each of them with their corresponding Distance and/or SetOperator.

Assume that the **StandardSetOperator** is the 'union' and the **StandardDistance** is 0.5. It can be stipulated that if the maximum weight lies between 0.9 and 1 then the distance should be 0.1 and the set operator the 'intersection', that if it lies between 0.8 and 0.9 then the distance should be 0.2 and the set operator the 'intersection' and that if it lies between 0.7 and 0.8 then the distance should be 0.3 (as nothing else is stipulated, the set operator to be applied in this interval is nothing else as the standard set operator).

Continuing with the above example but supposing that only the opinions of E_2 and E_3 exist then the distance which applies is 0.3 and the set operator for assigning a relationship to the "generalised edge" (PT_1,PT_2), will be the union, because 0.7 (the maximum value of both weights) lies between 0.7 and 0.8. Hence the assigned relationship will be $\{\equiv, \subset, \supset\}$.

Note that handling "generalised edges" is necessary every time the path-building algorithm encounters an edge whose nodes belong also to other edges. Assigning different values to the parameters defined above can thus greatly influence the output of the system.

The path issue

In order to increase efficiency some other restrictions besides break-off conditions can be relevant when building paths. The most important restriction regards the length of paths (**MaximalLength**). Finding out all paths existing between two potential taxa (nodes) can be very time consuming, because the algorithm will test every possible path no matter which length it reaches. Also, we presume intuitively that the longer a path the smaller the reliability of factual information transmission. Restricting the search to those paths whose length does not exceed a given boundary increases efficiency without significant loss of quality. This MaximalLength is a new break-off condition for building paths.

For the calculation of the path length we have assumed that all edges have the same length (actually a length of 1). This can be maintained as a general default, but it is useful to assign a particular **Length** to edges depending on the basic relationship they represent. In particular, if we pay attention to the changes in the applicability of factual information due to specific relationships (GEOFFROY & BERENDSOHN, 2003b), it makes sense to assign a length of 0 for edges with the 'congruency' relationship and / or a length greater than 1 for edges with the 'exclusion' relationship. We can thus on the one hand cluster potential taxa with identical concepts to form virtual nodes at least with respect to the MaximalLength parameter, and on the other hand put further distance between mutually excluding concepts.

Introducing weights for edges opens new perspectives for discarding some paths for further processing. However for this purpose we must first of all define a rule to assign a weight to a path on the basis of the edges involved in the path. In fact such a rule is a mathematical function. At least two reasonable directions can be taken: the first one is to take as rule the minimum over all edge weights, the second one is to take as rule the multiplication of all edge weights. This implies that a **StandardMathematical-Operation** should be another parameter in the configuration of the transmission engine.

If the multiplication operation is chosen, a more complex scenario could be implemented as well. E.g. a coefficient could be taken in account by the multiplication which would reflect somehow the length of the path. Take for instance a path P1,n = (PT1,…,PTn+1) with the edges E1 = (PT1,PT2), …., En = (PTn,PTn+1) and the respective weights W1,…,Wn; consider a coefficient for example of 0.9; we could define the weight W to be assigned to P1,n through $W = W1 * W2 * \ldots * Wn * (0.9)^n$.

Once weights can be assigned to paths, it is possible to break-off the process of building paths as soon as their weights fall under the ExcludeWeight, which was already defined for edges.

The generalised path issue

Since both edges and paths can be "simultaneous", and since a weight can be assigned to both, the considerations made for generalising edges with respect to discarding some edges in the calculation apply, in analogous form, for generalising paths. In contrast to the calculation for edges, for paths we do not need to base assessments on the weight of the source but only to the weights of paths as described above. The same set of parameters (the ExcludeWeight, the IntervalForMaximumWeight, the StandardSetOperator, the SetOperator and the Distance) that was defined for edges can be used for paths.

Edges are in fact special cases of paths and "generalised edges" are special cases of "generalised paths".

With appropriate tuning, the set of break-off conditions described above will allow retrieval of relevant paths to be considered between two arbitrary potential taxa for further calculation in reasonable time. Others parameters serve to precisely define the algorithm assigning a combined relationship to a generalised path, as demonstrated for generalised edges.

2. Tuning the output (transmission of factual information to the user)

Apart from the transmission of factual information from one node to another within the potential taxon graph the communication of results to the user must be discussed. We posited that a detailed specification cannot be developed in theory but needs to be an integral part of the actual implementation process (GEOFFROY & BERENDSOHN, 2003a). However, one of the aims formulated there was that in order to fully exploit the results correctly, end-users of the gathered information need to be informed of possible caveats caused by the transmission process. We have already stated that indiscriminate transmittance of every fact stemming from any source regardless of the calculated applicability would not be a desirable solution (GEOFFROY & BERENDSOHN, 2003b).

The practical question that arises in the context of implementation is: which factual information, and with which comment, should be accessible for which users? No general answer is available at this stage and therefore we should start by implementing mechanisms to tune output to the user interface. The output should depend on:

- The user group
- The applicability category of (transmitted) factual information
- The access restriction category
- The individual treatment of the factual information source
- The certainty of the factual information ('doubtful', see BERENDSOHN & al., 2003).

The importance of user roles

We ought to first differentiate users according to their functions within the information system: system managers, experts, other systems, and end-users. While system managers naturally need access to all functions and parts of the system, expert taxonomists may be restricted to handling the taxonomic editor and, within that function, to a certain taxonomic group. The question of interfacing with other systems cannot be generalised apart from perhaps providing an XML schema defining the data contents, which possibly can be output. The end-users accessing the system by means of a query interface are also a very diverse group; however, it should be possible to further differentiate by assigning them **UserRoles** according to their needs and their degree of scientific expertise. The latter is of particular importance in the context of formulating the output explaining caveats concerning the information transmitted, so the "user role" is a meaningful parameter to tune output. As an initial solution, we could distinguish between taxonomists, biologists and laymen as general groups, and introduce special purpose groups (e.g.: nature protection agency) to denote special roles addressing access restriction concerns.

The role of applicability categories

The applicability categories of factual information to potential taxa conform to a hierarchy from the lowest level 'not applicable' to the highest level 'fully applicable'. It does not make any sense to transmit to users factual information, which is 'not applicable'. But it might be reasonable to control output by setting a parameter for a "critical level", under which output of any factual information is denied. A **StandardApplicabilityExclusionLevel** applies to all users and sources. This can be overruled by an individual **ApplicabilityExclusionLevel** which may be defined for specific user roles and/or particular sources (if both are set, the more stringent one applies).

The role of access restrictions

Sources providing factual information may want to exclude some of their information being transmitted to everyone. For example, access to factual information could be restricted if it entails some special risk (e.g. medical information or information concerning the protection of species). An attribute for facts is added to hold the value for this parameter, the **AccessRestrictionLevel**. At least the following hierarchy of values should be possible: 'unrestricted', 'problematic' and 'restricted'. The level of a fact should be set to 'problematic' if the interpretation of its content requires previous specific knowledge and could mislead uninformed users. The value 'restricted' is to be used if the content of the fact should be kept unknown for some user groups because of legal or strategic reasons. We assume that users who should have access to 'restricted' facts are also able to interpret 'problematic' facts. With this it becomes possible to filter output depending on the user group, if a specific value of the **AccessLevel** had been associated to this user group. In this case factual information with a higher restriction as the one stipulated by the parameter will not be transmitted to members of that user group. In future a further differentiated access of a user group to facts depending on their category (e.g. access of taxonomists to classification-related data with high level and to conservation data with low level) might be taken in account. This would imply of course modifications in the below described "configuration extension of the Berlin Model".

Combining parameters and comments

In summary, we propose for output tuning a setting consisting of:

- A standard applicability exclusion level for the category of the transmitted factual information for all user groups and for all sources.
- Combinations of specific applicability exclusion levels and user roles for any particular access restriction. A comment for output should be assigned to every combination.
- Combinations of specific applicability exclusion levels and sources. A comment, which should be visible to the user, could be assigned to every combination here, too.

3. The configuration extension of the Berlin Model

To realise the above-discussed adjustments of rules and output, the parameters must be set, stored, and passed to the rules. The resulting conditions could be formulated in a formal language adapted for propositional calculus. Prolog (ROUSSEL, 1975) is one of the programming languages that fulfil these requirements. For the implementation in the context of the system here devised, however, rules could be implemented as stored procedures, and the needed parameters be read at runtime.

XML files are a valid option for managing and structuring such a configuration. For an example of such an XML configuration file and the corresponding XML schema see GEOFFROY (2002).

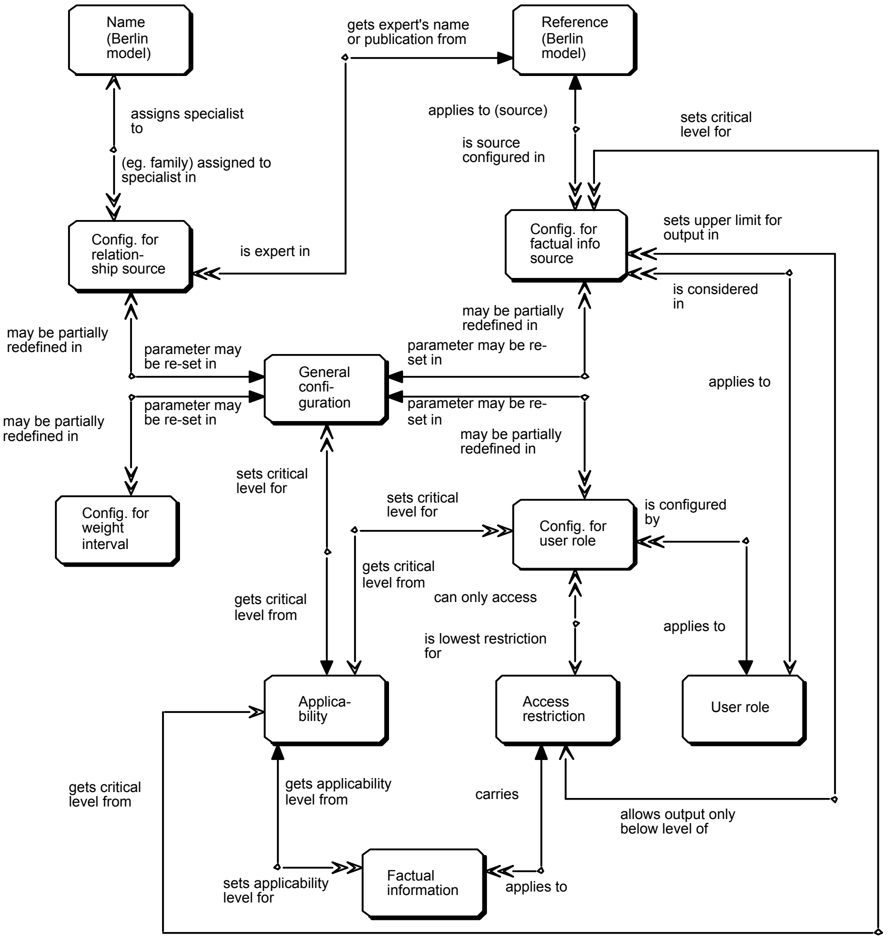

Name (Berlin model)
gets expert's name or publication from
Reference (Berlin model)
assigns specialist to
(eg. family) assigned to specialist in
applies to (source)
is source configured in
sets critical level for
Config. for relation-ship source
is expert in
Config. for factual info source
sets upper limit for output in
is considered in
may be partially redefined in
may be partially redefined in
parameter may be re-set in
General confi-guration
parameter may be re-set in
applies to
parameter may be re-set in
parameter may be re-set in
may be partially redefined in
may be partially redefined in
sets critical level for
sets critical level for
is configured by
Config. for weight interval
Config. for user role
gets critical level from
can only access
gets critical level from
applies to
is lowest restriction for
Applica-bility
Access restriction
User role
gets critical level from
gets applicability level from
carries
allows output only below level of
sets applicability level for
Factual information
applies to

Figure 1: ER-diagram of the model extension for the configuration module

An alternative to complex configuration files and the solution here pursued is to store parameters in tables within the core database. In the following, the entities required are described as an extension of the Berlin Model (BERENDSOHN & al., 2003). Figure 1 depicts the overall ER-Model for the extension.

The catalogue tables

Three new tables are required to represent the values accepted in the system for the parameters describing applicability, user role and access restriction to factual information. They have two attributes each, the primary key and the attribute corresponding to the table name.

The **APPLICABILITY** table holds the list of possible values for the applicability of factual information to potential taxa. Values for Applicability are, e.g. 'Fully applicable', 'Partially applicable', 'Doubtfully applicable', and 'Not applicable'.

The **USERROLE** table holds the list of possible values for the roles a user can have regarding the information system. Values for UserRole include 'System manager', 'Expert', 'Taxonomist', 'Biologist', and 'Layman'.

The **ACCESSRESTRICTION** table holds the list of possible values for the access restrictions to factual information, i.e. currently the values 'unrestricted', 'problematic', and 'restricted' for the attribute AccessRestriction.

The additional fact (MT_FACT) table

This table stores the information about the characterisation of factual information with respect to its original applicability (i.e. its applicability for the potential taxon to which it is linked at the source) and with respect to access restrictions imposed by the data provider. This information must be delivered directly from the data provider, if the factual database is accessed on-line. Otherwise the factual information itself is to be stored in the FACT table of the Berlin Model, but that table does allow storing of either applicability or access restriction. An additional table MT_FACT ("MT" for "MoReTax") is therefore required, which is connected by a 1-to-1 relationship with the FACT table. The MT_FACT table specifies for each fact its access restriction and its original applicability (Table 1).

Table 1: Attributes of the MT_FACT table

Short name	Type	Description
FactId	int	Primary key for table MT_FACT and one-to-one pointer to FACT
AccessRestrictionFk	int	"Access restriction level", pointer to ACCESSRESTRICTION
ApplicabilityFk	int	Pointer to APPLICABILITY

The general configuration table (CONFIGGENERAL)

A configuration is nothing else but a list of parameter values. The system should be able to store an unlimited number of different configurations, of which normally only one (the 'active' configuration") is applied to external queries. Former active configurations should not be erased but marked as 'not active'. Configurations, which are in the definition process, can be marked as 'provisional'.

A configuration must at least provide those global parameter values, which allow unambiguously applying the rules for the transmission (within the potential taxon graph as well as for the transmission of results to the user). The process should not depend on further detailed handling of particular experts, particular weight intervals, particular data provider or particular user groups. This basic set of parameters states how the weight of paths ought to be calculated, which paths are to be discarded and the minimal conditions for releasing factual information for output. If particular values are nowhere stipulated in the tables detailing the edges then a default length of 1 and a default weight of 1 will be assumed. The CONFIGGENERAL table holds these global parameters (Table 2).

Table 2: Attributes of the CONFIGGENERAL table

Short name	Type	Description
ConfigGeneralId	int	Primary key for table CONFIGGENERAL
ConfigStatus	str	Indicates whether the configuration is ‘active’, ‘provisional’ or ‘not active’
OperationForWeights	str	Numerical operator for weights when concatenating edges (e.g. multiplication)
StandardDistance	int	Maximal allowed distance between the maximum occurring weight and the weight of a path to be taken in account in case of simultaneous paths
ExcludeWeight	int	Minimal weight below which no path is taken in account
PathLength	int	Maximal allowed length of paths
LengthForCongruency	int	Length for an edge with the ‘congruent’ relationship
LengthForInclusion	int	Length for an edge with either ‘included in’ or ‘includes’ relationship
LengthForOverlap	int	Length for an edge with the ‘overlaps’ relationship
LengthForExclusion	int	Length for an edge with the ‘excludes’ relationship
StandardSetOperator	str	Set operator (intersection or union) for relationships in case of simultaneous paths
ApplicabilityFK	int	“Standard applicability exclusion level”, pointer to APPLICABILITY (indicates the applicability category below which output is not allowed)
ShowDoubtfulFactFlag	bool	Flag to indicate whether doubtful facts are allowed for output

The configuration detail tables

Each general configuration can be complemented with further details. For transmitting factual information across the potential taxon graph one of the crucial issues is the weight of edges. The **CONFIGRELSOURCE** table specifies for each source/expert who established relationships between potential taxa how these relationships and therefore the corresponding edges are to be weighted (Table 3).

Table 3: Attributes of the CONFIGRELSOURCE table

Short name	Type	Description
ConfigRelSourceId	int	Primary key for table CONFIGRELSOURCE
RelRefFK	int	Pointer to REFERENCE (a source responsible for relationships between potential taxa)
ExpertInTaxonFK	int	Pointer to NAME (family or genus the source is specialised in)
Weight	int	Standard weight for edges (relationships) created by the source
WeightForCongruency	int	Special weight for the 'congruent' relationship when created by the source
WeightForInclusion	int	Special weight for the 'included in' or the 'includes' relationship when created by the source
WeightForOverlap	int	Special weight for the 'overlaps' relationship when created by the source
WeightForExclusion	int	Special weight for the "exclusion" relationship when created by the source
ConfigGeneralFK	int	Pointer to CONFIGGENERAL

Note that in a configuration two records can exist for each source. One of them sets the weights for the taxonomic group in which the source is specialised and the other the weights outside this group. The REFERENCE and NAME entities both belong to the Berlin Model (BERENDSOHN & al., 2003).

Particular set operators and particular exclusion criteria for simultaneous paths different to those stipulated in the general configuration can be set for particular weight intervals (within which lies the maximum weight of simultaneous paths). These can be stored as special parameter values in the **CONFIGWEIGHT** table (Table 4):

Table 4: Attributes of the CONFIGWEIGHT table

Short name	Type	Description
ConfigGeneralFK	int	Pointer to table CONFIGGENERAL and part of the primary key
FromWeight	int	Lower limit of interval for maximum weight of simultaneous paths and part of the primary key
ToWeight	int	Upper limit of interval for maximum weight of simultaneous paths and part of the primary key
Distance	int	Maximal allowed distance between the maximum occurring weight lying within the interval and the weight of a path to be taken in account in case of simultaneous paths
SetOperator	str	Set operator (intersection or union) for relationships when the maximum weight lies within the interval

The output of query results (factual information) for the user depends not only on the general configuration but may also depend on the user role in combination with the access restriction for particular factual information and/or in combination with the particular factual information source. For any user role (and depending eventually on any particular access restriction) the transmission of the factual information can be tuned by the lowest applicability category allowed and a corresponding optional comment, as well as by allowing doubtful information to be taken into account or not. The **CONFIGUSER** table holds these parameters (Table 5).

Table 5: Attributes of the CONFIGUSER table

Short name	Type	Description
ConfigUserId	int	Primary key for table CONFIGUSER
UserRoleFK	int	Pointer to USERROLE
AccessRestrictionFK	int	Access level, points to table ACCESSRESTRICTION
ApplicabilityFk	int	Applicability exclusion level, points to table APPLICABILITY
OutputComment	text	Output comment for combination of user role, access restriction and applicability category
ShowDoubtfulFactFlag	bool	Flag to indicate whether doubtful facts are allowed for output
ConfigGeneralFK	int	Pointer to CONFIGGENERAL

This structure allows to control output for each user role, either for combinations of the two parameters - access restriction and applicability - or for one of both or even for neither of them. If for instance 'layman' users should never get doubtful information, then the attributes AccessRestrictionFK and ApplicabilityFk should be kept empty while the ShowDoubtfulFactFlag should be set to 'false'.

Similarly, for any factual information source (and depending eventually on any particular user role) the transmission of the factual information can also be regulated by a lowest applicability category allowed and a corresponding optional comment as well as by the setting about doubtful information. The **CONFIGFACTSOURCE** table holds these parameters (Table 6).

Table 6: Attributes of the CONFIGFACTSOURCE table

Short name	Type	Description
ConfigFactSourceId	int	Primary key for table CONFIGFACTSOURCE
SourceRefFK	int	Pointer to REFERENCE
UserRoleFK	int	Pointer to USERROLE
ApplicabilityFk	int	Applicability exclusion level, points to APPLICABILITY
OutputComment	text	Output comment for a combination of user role, fact source and applicability category
ShowDoubtfulFactFlag	bool	Flag to indicate whether doubtful facts are allowed for output
ConfigGeneralFK	int	Pointer to CONFIGGENERAL

For every factual information source output control can be set either for combinations of the two parameters – user role and applicability - or for one of both or even for neither of them. If for instance 'layman' users should be told that a particular source is not a scientific work whenever factual information has its origin in it, then the attribute ApplicabilityFk should be kept empty.

4. The rule tuner interface

The "rule tuner interface" is a program that allows administrators of the information system to configure the system over the World Wide Web. This remote editor tool requires on the client side only the presence of a web browser whereas dynamic html or shtml pages must be created on the server side (Figure 2).

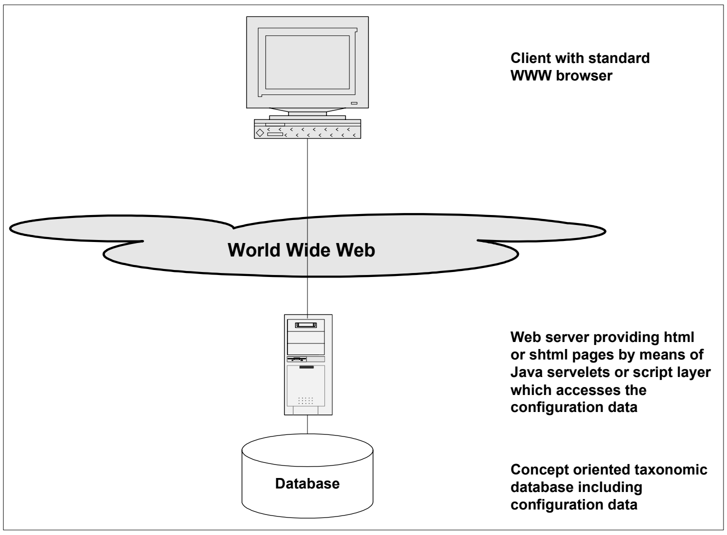

Client with standard WWW browser
World Wide Web
Web server providing html or shtml pages by means of Java servelets or script layer which accesses the configuration data
Database
Concept oriented taxonomic database including configuration data

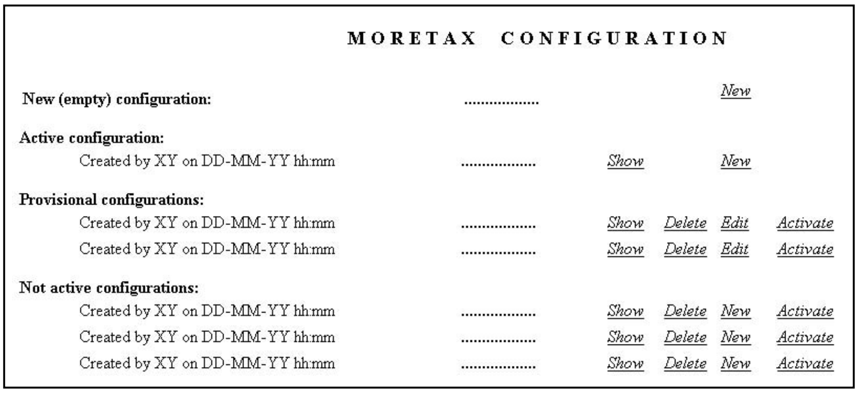
Figure2: Rule Tuner system architecture

The rule tuner interface on the web amounts to a set of (html-) forms, which enable the system manager to see parameter values corresponding to any configuration and to set parameter values for new (provisional) configurations, which are to be transferred to the server and to be stored in the configuration tables of the database. A set of example forms was developed as a first prototype to cover the functional needs of the system.

The main form allows to choose between existing configurations and gives access to the central configuration managing functions (Figure 3).

Its creation date and its author uniquely identify a configuration. Choices are:

- show the settings of an existing configuration
- delete a provisional or not active configuration (which only means that the configuration will not be listed any more in this form),
- create a new provisional configuration (either with no parameter values or based on an existing configuration by copying its parameter values for further editing).
- edit a provisional configuration
- activate a provisional or not active configuration (that means substituting the current active configuration, which then becomes inactive).

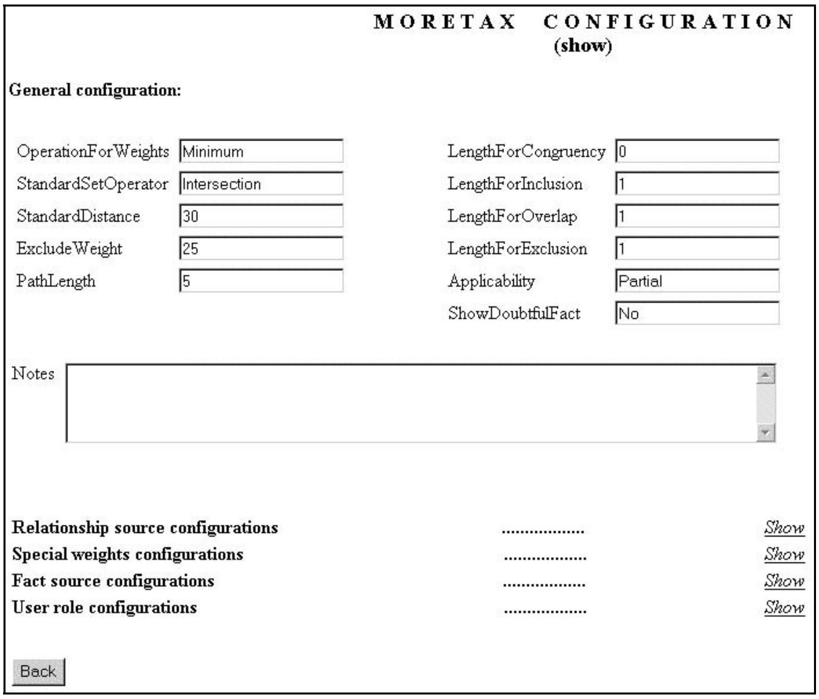


MORETAX CONFIGURATION

New (empty) configuration:	....			New	
Active configuration:					
Created by XY on DD-MM-YY hh:mm	....	Show		New	
Provisional configurations:					
Created by XY on DD-MM-YY hh:mm	....	Show	Delete	Edit	Activate
Created by XY on DD-MM-YY hh:mm	....	Show	Delete	Edit	Activate
Not active configurations:					
Created by XY on DD-MM-YY hh:mm	....	Show	Delete	New	Activate
Created by XY on DD-MM-YY hh:mm	....	Show	Delete	New	Activate
Created by XY on DD-MM-YY hh:mm	....	Show	Delete	New	Activate

Figure 3: The main configuration form: selecting a configuration to work on

The rule tuner interface will not be used for constant data input, but for the tweaking of individual settings and for experimentation with different sets of values. For that purpose, system managers can use a non-public user interface for which they can rapidly change the active configuration and use a provisional configuration to immediately control the effects of new settings. For the prototype forms, a strict separation between displaying the data ("Show" forms) and editing ("Edit" forms) was pursued. Direct editing is only possible for provisional configurations.



MORETAX CONFIGURATION
(show)

General configuration:

OperationForWeights	Minimum	LengthForCongruency	0
StandardSetOperator	Intersection	LengthForInclusion	1
StandardDistance	30	LengthForOverlap	1
ExcludeWeight	25	LengthForExclusion	1
PathLength	5	Applicability	Partial
		ShowDoubtfulFact	No

Notes

Relationship source configurations	....	Show
Special weights configurations	....	Show
Fact source configurations	....	Show
User role configurations	....	Show

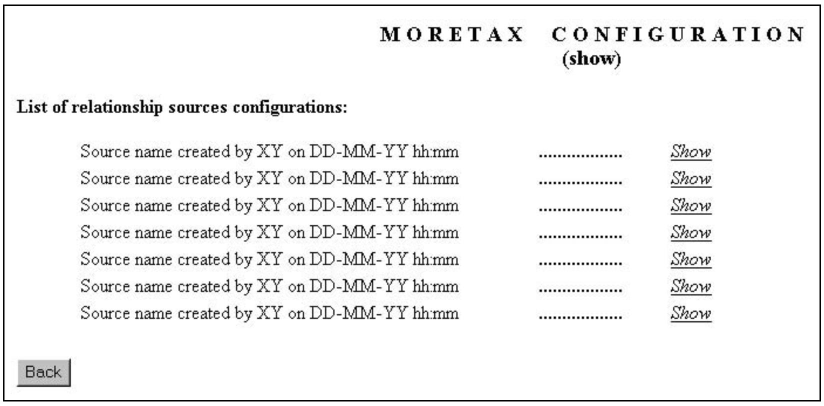
Figure 4: Display of the general configuration

Displaying the configuration: the “Show” forms

The first form to appear after selecting the “Show” link (Figure 3) lists the parameters from the ConfigGeneral table (Figure 4) of the database and makes it possible to show further details of the chosen configuration (data from the related tables ConfigRelSource, ConfigWeight, ConfigFactSource and ConfigUser).

Weights and distances are here represented as percent values, so that 75 actually means 0.75.

The form showing the concept relationship’s source configurations (Figure 5) contains the list of records of the ConfigRelSource table (i.e. the sources or authors who have set the relationships between potential taxa). Each record can be distinguished through the name of the source and its creation data.



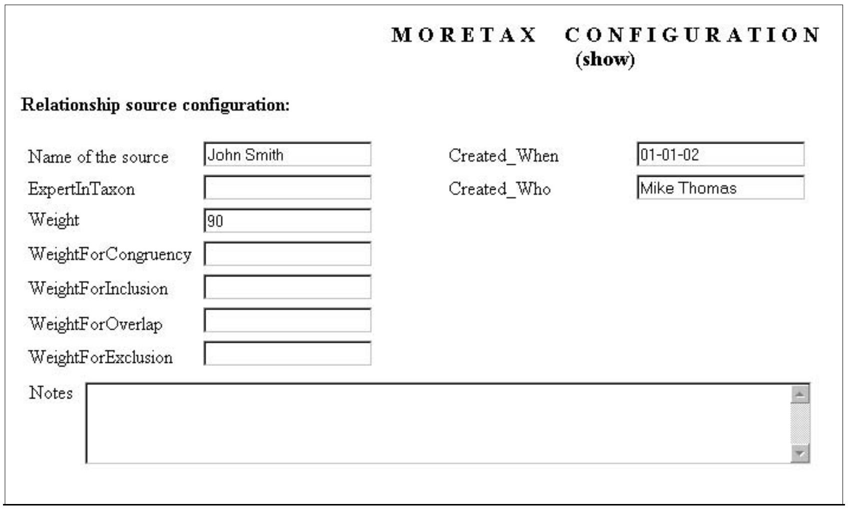
MORETAX CONFIGURATION
(show)

List of relationship sources configurations:

Source name created by XY on DD-MM-YY hh:mm .... Show
Source name created by XY on DD-MM-YY hh:mm .... Show
Source name created by XY on DD-MM-YY hh:mm .... Show
Source name created by XY on DD-MM-YY hh:mm .... Show
Source name created by XY on DD-MM-YY hh:mm .... Show
Source name created by XY on DD-MM-YY hh:mm .... Show
Source name created by XY on DD-MM-YY hh:mm .... Show

Figure 5: Listing and choosing relationship sources

The form shown in Figure 6 contains the significant fields and their values for the chosen record from the ConfigRelSource table.



MORETAX CONFIGURATION
(show)

Relationship source configuration:

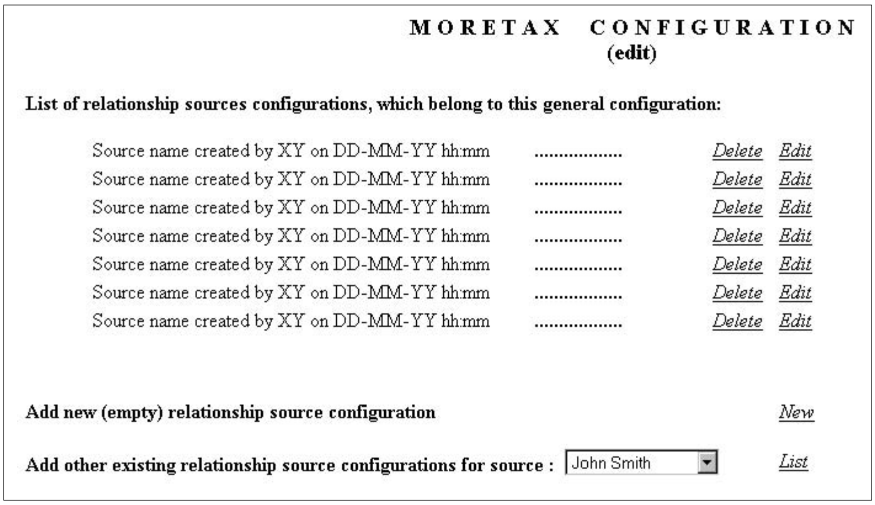
Name of the source	John Smith	Created_When	01-01-02
ExpertInTaxon		Created_Who	Mike Thomas
Weight	90		
WeightForCongruency			
WeightForInclusion			
WeightForOverlap			
WeightForExclusion			
Notes			

Figure 6: The form to display individual relationship sources and their weights

Similar forms are needed to show the content of the CONFIGWEIGHT, CONFIGFACTSOURCE, and CONFIGUSER tables (see Figures 8, 9, and 10 for the editable analogues). Where special weights are assigned, the forms show both, the default parameters (the same as in the "Show Configuration Form") and the list of records from the CONFIGWEIGHT table for this general configuration. Each record can be recognised through the particular weight interval it deals with. The form showing the details of a particular weight interval and the respective parameters should also show the values for the general configuration. Similarly, in the case of fact source configurations and of user role configurations, the detail forms should always contain the relevant values for output already set in the general configuration. Here, individual records listed can be distinguished by means of the values for source and user role in the case of the fact source configurations, and by user role and access restriction in the case of the user role configurations.

Changing entries: the "Edit" forms

Edit forms, accessible through the "Configuration Main Form" (Figure 3), are only available for provisional configurations. They allow system managers to modify (and not only to view) the configuration either by altering values within records or by adding or deleting records in the CONFIGRELSOURCE, CONFIGWEIGHT, CONFIGFACTSOURCE or CONFIGUSER tables. The structure of these forms is of course very similar to the "Show" forms.



MORETAX CONFIGURATION
(edit)

List of relationship sources configurations, which belong to this general configuration:

Source name created by XY on DD-MM-YY hh:mm .... *Delete* *Edit*
Source name created by XY on DD-MM-YY hh:mm .... *Delete* *Edit*
Source name created by XY on DD-MM-YY hh:mm .... *Delete* *Edit*
Source name created by XY on DD-MM-YY hh:mm .... *Delete* *Edit*
Source name created by XY on DD-MM-YY hh:mm .... *Delete* *Edit*
Source name created by XY on DD-MM-YY hh:mm .... *Delete* *Edit*
Source name created by XY on DD-MM-YY hh:mm .... *Delete* *Edit*

Add new (empty) relationship source configuration *New*

Add other existing relationship source configurations for source : John Smith *List*

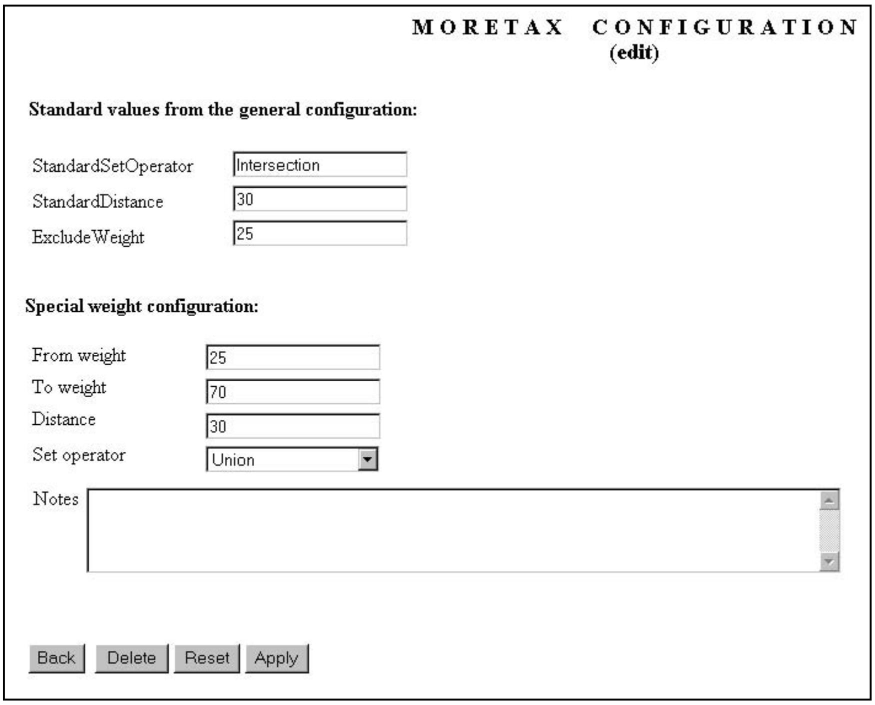
Figure 7: Choosing relationship source configurations for editing

The first form to appear (the "Edit Configuration Form") is almost identical to the "Show Configuration Form" (Figure 4). It lists, for edition, the parameters from the chosen general configuration and allows picking out further details of the configuration to be edited. Apart from the browser's standard "Back" button, which affects neither the entries made nor the entries in the database, a "Reset" button cancels the intended modifications and re-displays the values from the database, while an "Apply" button commits the changes made to the actual database record.

The choice to list the relationship source configurations calls a form (Figure 7) which is similar to the "List Relationship Sources" form (Figure 5), but for the possibilities

- to delete (= set all "weights" to 0) or edit any relationship source configuration already associated to the provisional configuration and
- to add new relationship source configurations to the provisional configuration either by creating a new relationship source configuration or by associating to the selected provisional general configuration an existing relationship source configuration for one of the sources existing in the entire system.

Note that this configuration actually chooses the relationship sources to be taken into account – the default weight for a relationship source is 0, so if no configuration is set, the corresponding edges of the potential taxon graph will not normally be considered (at least if the exclude weight is defined as being > 0).



MORETAX CONFIGURATION
(edit)

Standard values from the general configuration:

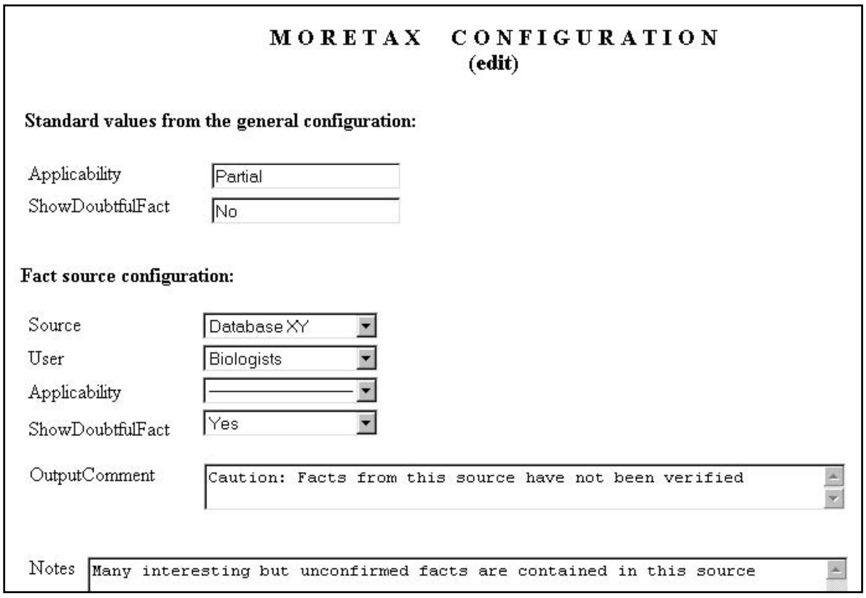
StandardSetOperator: Intersection
StandardDistance: 30
ExcludeWeight: 25

Special weight configuration:

From weight: 25
To weight: 70
Distance: 30
Set operator: Union
Notes:

Figure 8: The form to edit the special weights configuration

Adding a source to the configuration is a three-step process. First a source reference is selected (only those that have not been configured for this general configuration are shown), and then a list of all existing configurations for that relationship source is generated. This list is displayed in a form identical to the one showing the list of relationship sources (Figure 5), except that the "Show" link is replaced by an "Edit" link, and, alternatively, that a new (empty) configuration can be created for the source. Both options then lead to a form for editing (a copy of) the chosen record, which is now associated with the selected provisional configuration. This form is similar to the one shown in Figure 6. The data on the creation serve as background information and should therefore not be editable.



MORETAX CONFIGURATION
(edit)

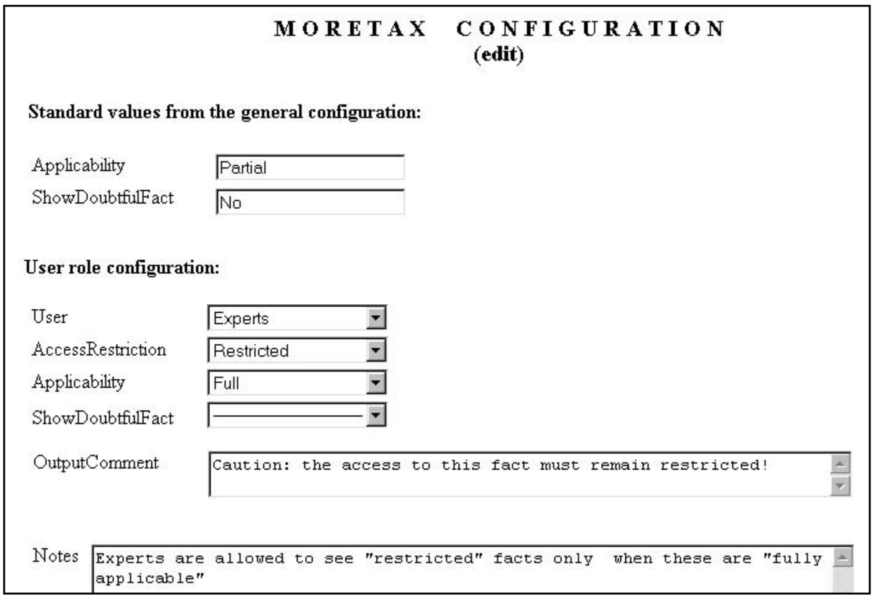
Standard values from the general configuration:

Applicability	Partial
ShowDoubtfulFact	No

Fact source configuration:

Source	Database XY
User	Biologists
Applicability	――――――
ShowDoubtfulFact	Yes
OutputComment	Caution: Facts from this source have not been verified
Notes	Many interesting but unconfirmed facts are contained in this source

Figure 9: The form to edit fact source configurations (buttons below as in Figure 8)



MORETAX CONFIGURATION
(edit)

Standard values from the general configuration:

Applicability	Partial
ShowDoubtfulFact	No

User role configuration:

User	Experts
AccessRestriction	Restricted
Applicability	Full
ShowDoubtfulFact	――――――
OutputComment	Caution: the access to this fact must remain restricted!
Notes	Experts are allowed to see "restricted" facts only when these are "fully applicable"

Figure 10: The form to edit user role configurations (buttons below as in Figure 8)

The steps just detailed for relationship source configurations apply in an analogous way to the configurations for special weights, fact sources or user roles selected from the edit form of the general configuration (Figure 4). The corresponding ‘Edit’ link allows deleting, adding, or editing detail configurations (see Figures 8, 9, and 10).

5. The database interfaces

Database interfaces are software components enabling the communication between the core database and factual databases. We assume that this communication occurs through the World Wide Web (GEOFFROY & BERENDSOHN, 2003a). The transmission engine distributes user queries against the factual databases. The data providers play the role of servers whereas the transmission engine here officiates as client, which dynamically creates the queries to be transferred to the servers. Following actual technology trends, and in order to be independent from operating systems, queries will be XML encoded and send by means of an http-request (e.g. “post”). A so-called "wrapper" (a piece of software, which translates the XML encoded queries into the native query language of the factual data server – e.g. as SQL) intercepts these XML queries on the server side.

If wrappers are to be independent from the respective structure of the databases with factual information, then there ought to be common views against which the queries will be formulated. This means that all databases with factual information must make at least part of their data available through such standardised views.

The wrappers are also responsible for returning the query results (e.g. the content of record sets) embedded into an XML document to the client (transmission engine). There it will be parsed, integrated, and, if needed, transformed into the user interface format (e.g. html). Figure 11 illustrates these database relations.

Apart from this interface for factual data, another database interface is used to handle the names and taxonomic data to be retrieved from factual databases for incorporation in the core database. Both interfaces will be discussed in the following sections.

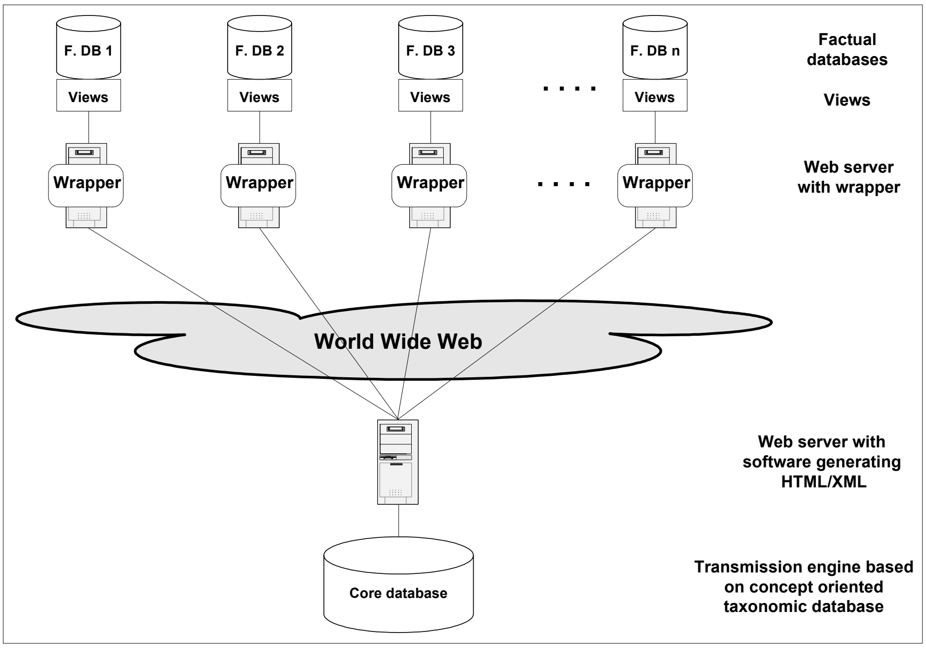


Figure 11: Communication between databases

The database interface for nomenclatural and taxonomic data import and update

Taxon names used in factual databases must be present as taxonyms in the core database in order to allow experts to establish concept relationships. As a side effect, this also increases the efficiency of the system, because databases will subsequently not be queried for names for which they do not hold information. Taxon names used in the factual database but ascribed in it to other source or circumscription references should also be present in the core database (as taxonyms).

Updating nomenclatural and taxonomical data, which affect taxonomic concepts, should not happen too frequently, since experts must establish new relationships before these updated data can be effectively used by the information system.

Consequently, an interface is needed for importing and updating data from the databases containing factual information. This interface involves:

- the taxon names domain
- the nomenclature references domain
- and eventually the “potential taxa” references domain.

For this purpose we need a large view on factual databases. In order to suit possible different database structures this view must consider data in an atomised and also in a not atomised form and must enable owners of each factual database to provide the data as detailed as they want or can. It should be possible to offer further information with some notes fields and to check by means of date fields whether the same data have already been imported. The fields of this view refer to the three domains we mentioned above. Fields of the third domain will only be filled if potential taxa explicitly exist in the factual database, otherwise the empty fields in this domain will be so interpreted that the factual database itself will be used as the reference. The view consists primarily of attributes already defined for the Berlin Model (BERENDSOHN & al., 2003):

- All attributes from the table NAME are included, but the primary key attribute is to be used here as a unique identifier for the taxonym within the source. An attribute AuthorTeamString for the complete author citation and an attribute NomStatus for the nomenclatural status have to be added to cover the taxon names domain.
- To cover the domain of nomenclatural references, all attributes from the table REFERENCE are included except for the foreign keys and the attribute URL; and additionally, the attribute FullNomRefCache from the table REFDETAIL.
- If potential taxa exist in the source, they can be addressed by the attributes in table PTAXON excluding IdInSource and the foreign keys except StatusFk. Apart from that, an attribute AcceptedTaxonymFk is included in the view as a pointer to the accepted taxonym (used only if the status is synonym), and an attribute HigherTaxonFk points to the higher-ranking taxonym to designate the hierarchical classification (only for taxa accepted by the source). In addition, we need FullNomRefCache as defined in the table REFDETAIL, and the attributes from the table REFERENCE without the foreign keys.

Retrieving factual information

End-users can query the system for the factual information once the taxon names are imported from the factual database and the corresponding potential taxa are integrated in the potential taxon graph using implicit relationships (in combination with the appropriate configuration settings) or relationships explicitly established by experts. In order to retrieve it from each factual database, another view is needed.

Table 10: The preliminary view for factual information

Field name	Type	Field description
FactId	int	Unique identifier for the record
Fact		Placeholder for factual information linked to the potential taxon
FactCategory	str	Category of the factual information (e.g. 'natural product content ')
AccessRestriction	str	Access restriction for the factual information (default is 'unrestricted')
PtaxonFk	int	Unique identifier of the potential taxon the fact is linked to
Applicability	str	Applicability of factual information (default is 'fully applicable') to potential taxon
ConceptChangedFlag	bool	Set if the source's concept about the taxon has changed
Created_When	date	Date and time when factual information was created/updated
Notes	str	Remarks and notes for further details on factual information

This view should not only provide the facts, but also an indication if the concept in the source database has changed. In the view defined in Table 10 we tried to account for these requirements. However, as long as no concrete application involving real factual information sources is developed this remains a highly hypothetical endeavor. Information structure research is required to test the hypothesis that a single view can be construed to cover all kinds of factual databases. For the time being, the attribute Fact can either hold simple strings or numbers, or it is used as a placeholder for complex structured factual information.

6. The end-user interface

The "user interface" is a server-sided software program which enables the communication between end-users and the transmission engine (server). The user-sided client is in this case a standard web browser and the transmission engine the server. Output dynamically generated by the web server allows users to formulate queries against the transmission engine and to get back the results of such queries (Figure 12).

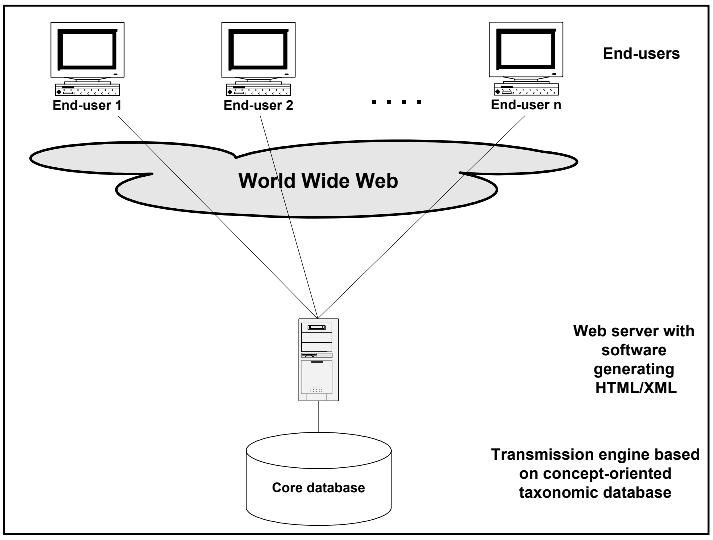
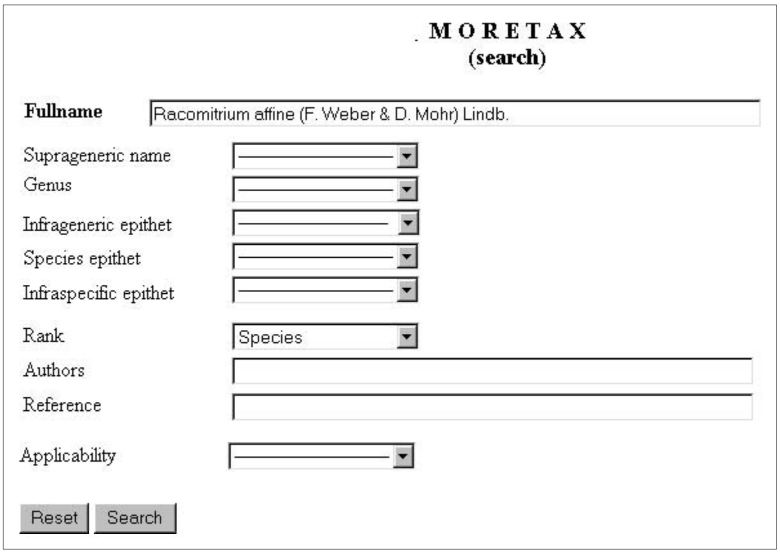

End-users
End-user 1
End-user 2
. . . .
End-user n
World Wide Web
Web server with software generating HTML/XML
Core database
Transmission engine based on concept-oriented taxonomic database

Figure 12: Communication between end-users and the core database

The forms we are going to describe are restrained to basic functionalities of the system. In fact, we expect that users will often start with one of the factual databases and then search for related information, or actually search for certain facts, instead of using the straight name-based query here illustrated. Also, because of the complexity of the information, a visualisation approach may be extremely helpful to navigate the overlapping hierarchies involved, as demonstrated by GRAHAM & al. (2000). Careful analysis of the user requirements is needed to implement any particular user interface. However, some features of the input and output functions can be demonstrated using this approach.

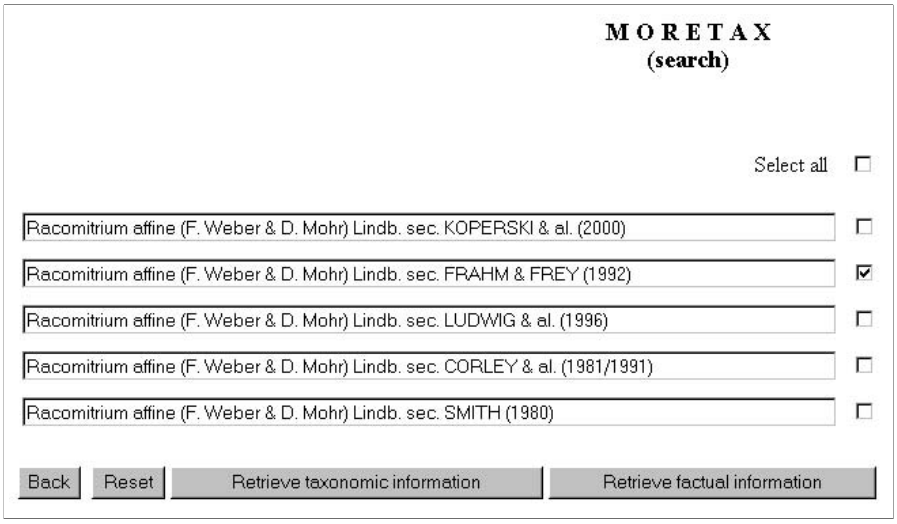
The user interface is presented as a sequence of forms, starting with a log-in form. Only the lowest security clearance can access the system without entering a password. Specialised users (e.g. experts) must be previously registered to be assigned to a group and so can be served appropriately. This means that it should be possible to access a "registration" form from the "log-in" where a new user can apply for a certain status in the system.



Fullname Racomitrium affine (F. Weber & D. Mohr) Lindb.

Rank Species

Figure 13: The taxon-centric central search form



Select all □

Racomitrium affine (F. Weber & D. Mohr) Lindb. sec. KOPERSKI & al. (2000) □

Racomitrium affine (F. Weber & D. Mohr) Lindb. sec. FRAHM & FREY (1992) ☑

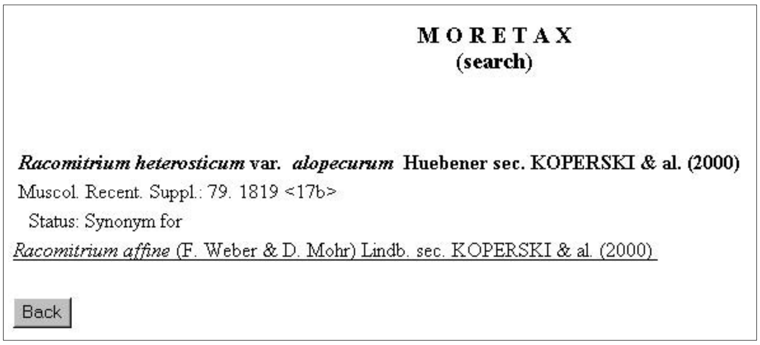
Racomitrium affine (F. Weber & D. Mohr) Lindb. sec. LUDWIG & al. (1996) □

Racomitrium affine (F. Weber & D. Mohr) Lindb. sec. CORLEY & al. (1981/1991) □

Racomitrium affine (F. Weber & D. Mohr) Lindb. sec. SMITH (1980) □

Figure 14: Potential taxa found for the name queried and selection of sources

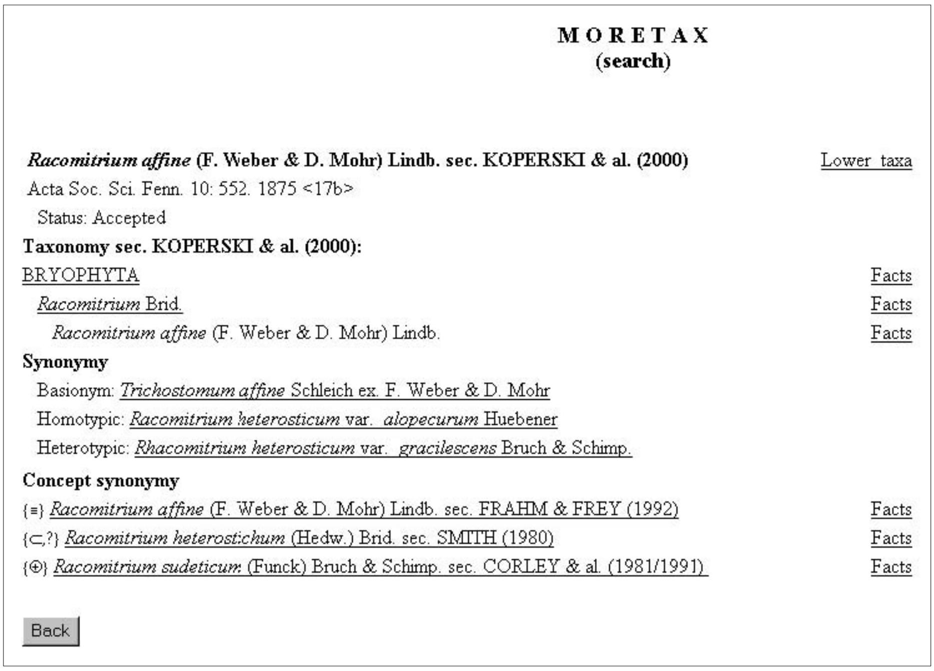
Once logged in, the user gets a basic search form (Figure 13), in which taxon names (atomised or not atomised) eventually with authors and (circumscription) references can be input for searching. Here the user also chooses the lowest applicability category for factual information output.



MORETAX
(search)

***Racomitrium heterosticum* var. *alopecurum* Huebener sec. KOPERSKI & al. (2000)**
Muscol. Recent. Suppl.: 79. 1819 <17b>
Status: Synonym for
Racomitrium affine (F. Weber & D. Mohr) Lindb. sec. KOPERSKI & al. (2000)

Figure 15: Taxonomic information output for synonymous taxonym



MORETAX
(search)

***Racomitrium affine* (F. Weber & D. Mohr) Lindb. sec. KOPERSKI & al. (2000)** Lower taxa
Acta Soc. Sci. Fenn. 10: 552. 1875 <17b>
Status: Accepted

Taxonomy sec. KOPERSKI & al. (2000):
BRYOPHYTA Facts
Racomitrium Brid. Facts
Racomitrium affine (F. Weber & D. Mohr) Lindb. Facts

Synonymy
Basionym: *Trichostomum affine* Schleich ex. F. Weber & D. Mohr
Homotypic: *Racomitrium heterosticum* var. *alopecurum* Huebener
Heterotypic: *Rhacomitrium heterosticum* var. *gracilescens* Bruch & Schimp.

Concept synonymy
{≡} *Racomitrium affine* (F. Weber & D. Mohr) Lindb. sec. FRAHM & FREY (1992) Facts
{⊂,?} *Racomitrium heterostichum* (Hedw.) Brid. sec. SMITH (1980) Facts
{⊕} *Racomitrium sudeticum* (Funck) Bruch & Schimp. sec. CORLEY & al. (1981/1991) Facts

Figure 16: Taxonomic information output for accepted taxonym

Executing the query leads to a new form (Figure 14) listing the potential taxa found under the queried name as retrieved from the taxonomic core database. The user can select one, several or all of them and then let the system search either for taxonomic information or for factual information corresponding to the selected items, as far as it is applicable. A 'back' button enables the user in all search forms to go back to the central search form.

Retrieving taxonomic information

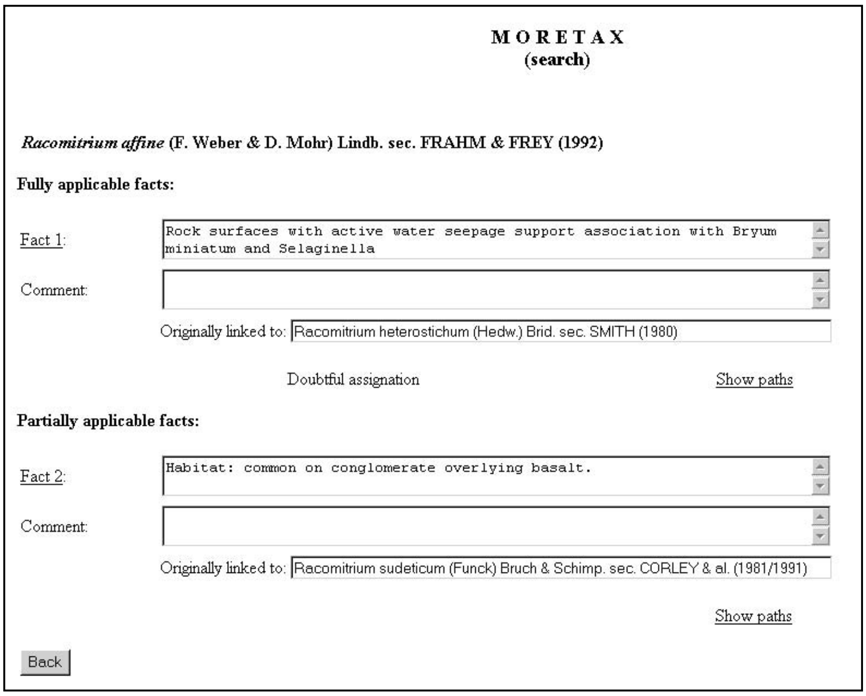
The form depicted in Figure 14 enables the user to retrieve the taxonomic information corresponding to the selected taxonyms from the core database. The results of that new query differ according to the status assigned between those taxonyms, which are considered as accepted (correct) by the source, and those that are not. In both cases, the taxonym (name, name authors, source "sec." reference) is cited, followed by the nomenclatural reference citation for the name and the status assigned to the name in the source. For synonyms (Figure 15), the only other item cited is the accepted (correct) name for which they are a synonym. For accepted names (Figure 16), more details are given instead:

- Classification branch from the highest rank up to the potential taxon (this classification is "subjective" in the sense that it represents the opinion of the source)
- (Traditional) synonymy according to the source reference
- Included taxa of lower rank according to the source reference
- Concept synonymy as established by relationship (expert) sources

From both forms, the user can choose any one of the taxonyms retrieved in order to get either the respective taxonomic information or the corresponding factual information.

Retrieving factual information and potential taxa

Querying for factual information (see Figure 14, fact categories are ignored in these examples) results in a list of all transmitted facts for each of the selected potential taxa,



MORETAX
(search)

Racomitrium affine **(F. Weber & D. Mohr) Lindb. sec. FRAHM & FREY (1992)**

Fully applicable facts:

Fact 1: Rock surfaces with active water seepage support association with Bryum miniatum and Selaginella

Comment:

Originally linked to: Racomitrium heterostichum (Hedw.) Brid. sec. SMITH (1980)

Doubtful assignation

Partially applicable facts:

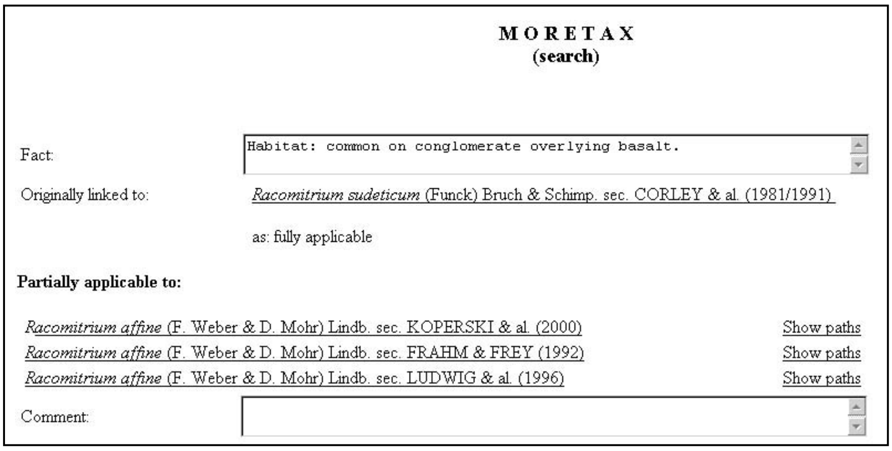
Fact 2: Habitat: common on conglomerate overlying basalt.

Comment:

Originally linked to: Racomitrium sudeticum (Funck) Bruch & Schimp. sec. CORLEY & al. (1981/1991)

Figure 17: Fact output and search form

sorted by the applicability category. Only facts fulfilling the minimal applicability criterion set at the outset are shown. The example cites facts as text strings, but as we already pointed out, much more complex structures are possible once an implementation with concrete factual databases is assembled. For each fact listed here, the output includes a comment, an indication of the potential taxon to which it was originally linked, and eventually the information that this last assignation was doubtful (if so) (Figure 17).



MORETAX
(search)

Fact: Habitat: common on conglomerate overlying basalt.

Originally linked to: *Racomitrium sudeticum* (Funck) Bruch & Schimp. sec. CORLEY & al. (1981/1991)

as: fully applicable

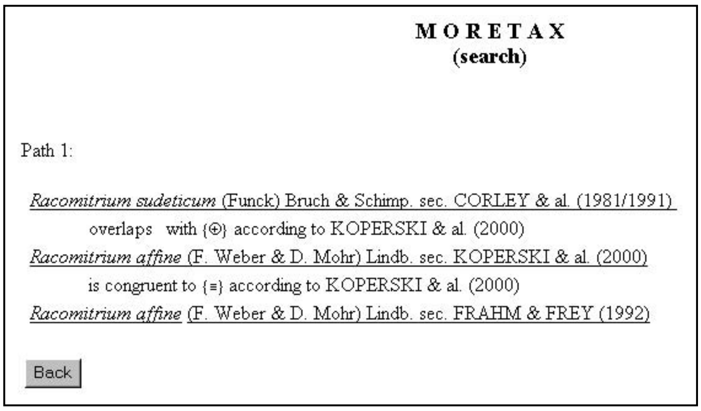
Partially applicable to:

Racomitrium affine (F. Weber & D. Mohr) Lindb. sec. KOPERSKI & al. (2000) Show paths
Racomitrium affine (F. Weber & D. Mohr) Lindb. sec. FRAHM & FREY (1992) Show paths
Racomitrium affine (F. Weber & D. Mohr) Lindb. sec. LUDWIG & al. (1996) Show paths

Comment:

Figure 18: Result of the query for other potential taxon to which the fact applies

From here, it should be possible to query the system about potential taxa to which a fact picked out is transmitted (the Fact # - link). The results are presented in Figure 18, which displays some meta-information about the selected fact, such as the potential taxon to which it was originally linked, the original applicability category, as well as the doubtful assignment indication.



MORETAX
(search)

Path 1:

Racomitrium sudeticum (Funck) Bruch & Schimp. sec. CORLEY & al. (1981/1991)
overlaps with {⊕} according to KOPERSKI & al. (2000)
Racomitrium affine (F. Weber & D. Mohr) Lindb. sec. KOPERSKI & al. (2000)
is congruent to {≡} according to KOPERSKI & al. (2000)
Racomitrium affine (F. Weber & D. Mohr) Lindb. sec. FRAHM & FREY (1992)

Figure 19: Result of the query for a path

The retrieved potential taxa should be listed according to the applicability categories of the factual information when transmitted to them. The comments for facts need only appear once for each combination of applicability category and source (comments depend on the user role, on the factual information source and on the applicability category).

In both forms the user can request information about the transmission of information from the system. The “Show paths” link results in a form describing all paths within the potential taxon graph that connect the selected potential taxon with the potential taxon the factual information was originally linked to. Figure 19 sketches the structure of a new form for paths description including not only the sequence of nodes but also the relationships ascribed to the edges and their authors. In the case of generalised edges, a link (‘Details’) is displayed instead of the author, leading to a form with the complete information.

Selecting any listed potential taxon either in this form or in the one in Figure 18 is interpreted as a new query about all factual information that can be transmitted to it. As a result, the user receives output as depicted in Figure 17.

Providing users with tools to get more accurate and commented factual information is the very goal of the theoretical approach we have dedicated our attention to. As mentioned before, concrete applications are now needed to evaluate the theoretical considerations addressed in this volume as well as to develop and fine-tune a user interface matching the requirements of concrete users.

References cited

BERENDSOHN, W. G., DÖRING, M., GEOFFROY, M., GLÜCK, K., GÜNTSCH, A., HAHN, A., KUSBER, W.-H., LI, J.-L., RÖPERT, D. & SPECHT, F. (2003): The Berlin Taxonomic Information Model. Schriftenreihe Vegetationsk. 39: 15-42.

GEOFFROY, M. (2002 [Jan 2, 2003]): Definition of semantics for serial relationships and formulation of a rule catalogue. http://www.bgbm.org/BioDivInf/Projects/MoreTax/Inference_rules_and_rule_adjustments.htm.

GEOFFROY, M. & BERENDSOHN, W. G. (2003a): The concept problem in taxonomy: importance, components, approaches. Schriftenreihe Vegetationsk. 39: 5-14.

GEOFFROY, M. & BERENDSOHN, W. G. (2003b): Transmission of taxon-related factual information. Schriftenreihe Vegetationsk. 39: 83-86.

GEOFFROY, M. & GÜNTSCH, A. (2003): Assembling and navigating the potential taxon graph. Schriftenreihe Vegetationsk. 39: 71-82.

GÜNTSCH, A., GEOFFROY, M., DÖRING, M., GLÜCK, K., LI, J.-J., RÖPERT, D., SPECHT, F. & BERENDSOHN, W. G. (2003): The taxonomic editor. Schriftenreihe Vegetationsk. 39: 43-56.

GRAHAM, M., KENNEDY, J. B. & HAND, C. (2000): A comparison of set-based and graph-based visualisations of overlapping classification hierarchies. Pp. 41-50 in: DI GESU, V., LEVIALDI, S. & TARANTINO, L. (eds.): Advanced Visual Interfaces, international working conference, Palermo, Italy. IEEE Computer Society Publishers.

ROUSSEL, P. (1975): Prolog : Manuel de Reference et d'Utilisation, Groupe d'Intelligence Artificielle, Faculté des Sciences de Luminy, Université Aix-Marseille II

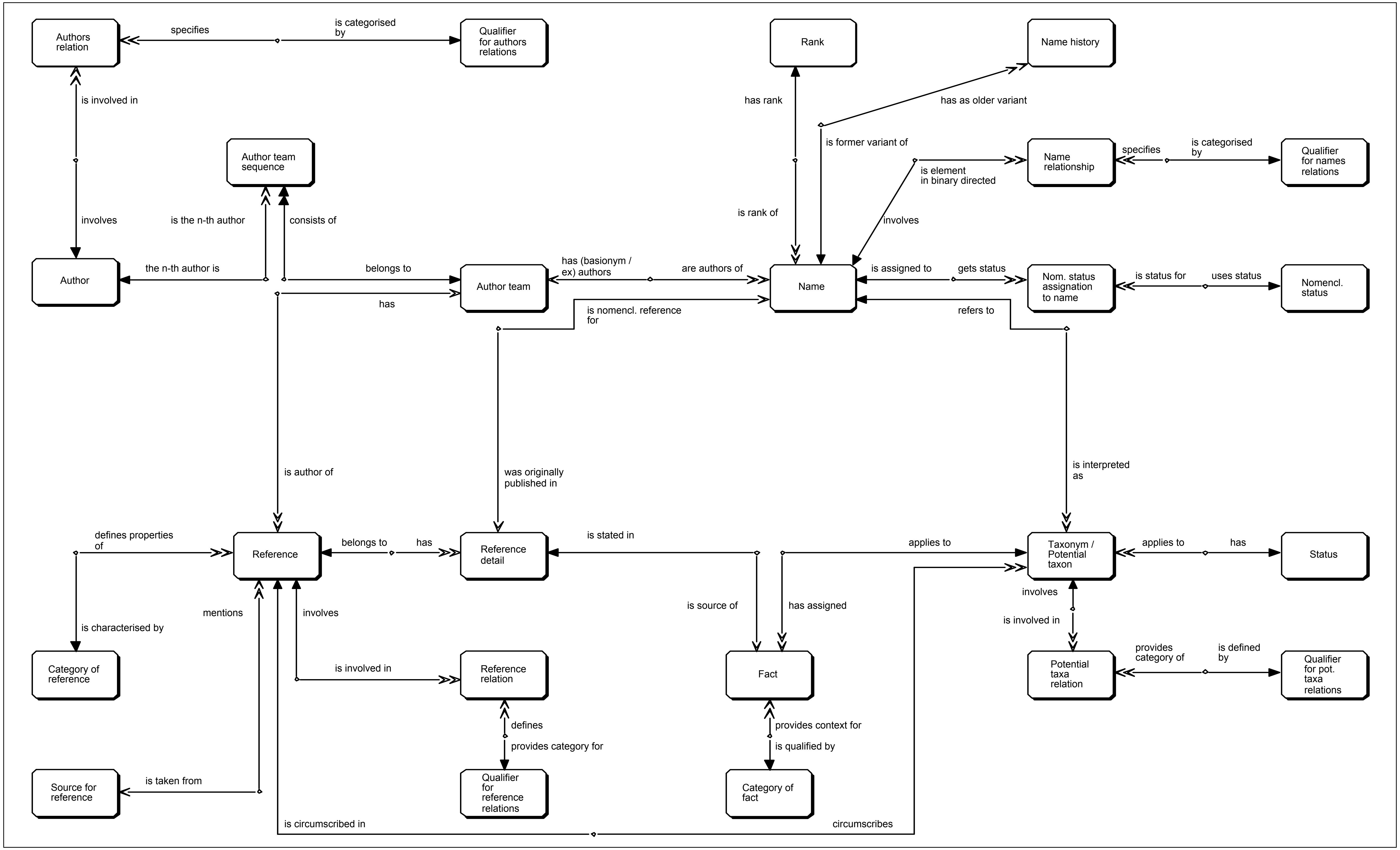
Acknowledgements

We should like to thank the members of MoReTax' accompanying working group and other participants in the discussions during the two sessions of the group for their most valuable input:

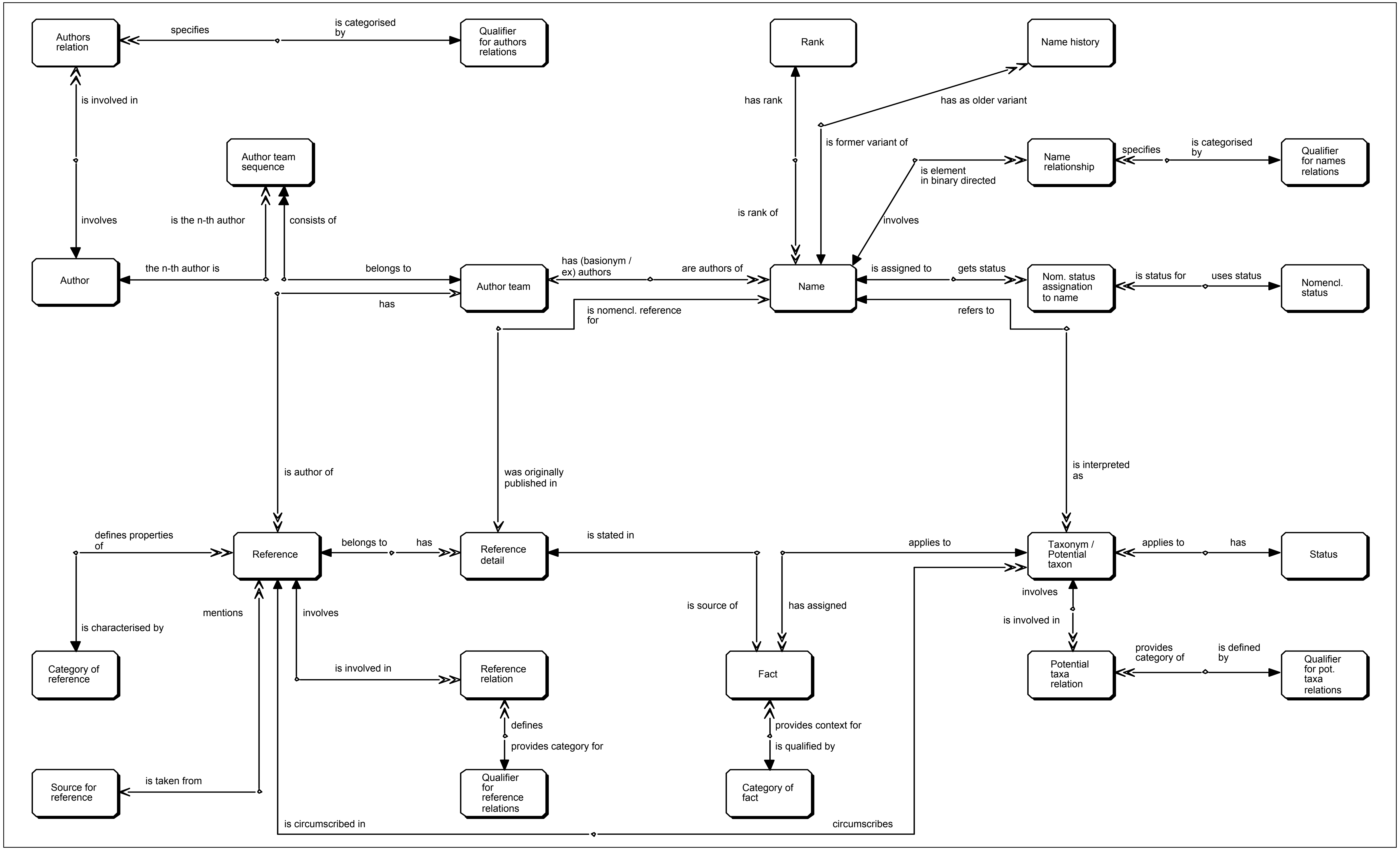
Walther Braun, ZADI - Zentralstelle für Agrardokumentation und –information, Bonn
Hans Fink, Bundesamt für Naturschutz, Bonn
Gregor Hagedorn, Federal Research Center for Agriculture and Forestry (BBA). Berlin
Norbert Hirneisen, Verlag für interaktive Medien, Gaggenau
Jürgen R.Hoppe, Abteilung Spezielle Botanik, Universität Ulm
Jacques Le Renard, Museum nat. d'Histoire naturelle, Paris
Rudolf May, Bundesamt für Naturschutz, Bonn
Sabine Roscher, ZADI - Zentralstelle für Agrardokumentation und –information, Bonn
Uwe Schippmann, Bundesamt für Naturschutz, Bonn
Mark F.Watson, Royal Botanic Garden, Edinburgh
Gisela Weber, ZADI - Zentralstelle für Agrardokumentation und –information, Bonn

Special thanks for native-speaker support of proofreading goes to Mark Watson (first and the four last articles), Karen Wilson of the RBG Sydney and Gregory C. Tjepkema (second article), and Steven Jury (University of Reading, third article).

The German Federal Environment Ministry through the Federal Agency for Nature Conservation funded the MoReTax project as well as the project that led to the Checklist of German Mosses. We should like to thank Hans Fink and Rudolf May of the Federal Agency for their continuous support and far-sighted comprehension of our research objectives.



ER-diagram of the Berlin Model (simplified)



ER-diagram of the Berlin Model (simplified)