

4D4Life



Distributed Dynamic Diversity Databases for Life

Deliverable 7.1A

Requirements, Specification and Design of the e-2 architecture

Work package 7

Richard White, Jonathan Giddy, Andrew
Jones, Alex Hardisty, Hardik Raja

30 April 2010

Capacities Programme of Framework 7: EC e-Infrastructure Programme -
Scientific Data Infrastructures INFRA-2008-1.2.2

Grant Agreement No:	238988
Project Co-ordinator:	Professor Frank Bisby
Project Homepage:	http://www.4D4Life.eu
Duration of Project:	36 months
Start Date:	01 May 2009
End Date:	30 May 2012

Report on Requirements, Specification and Design of the new e-2 architecture

Status

4D4Life Deliverable 7.1A version 1.1, 30 April 2010

An updated version (Deliverable 7.1B) will follow by 31 August 2010. It will continue to be updated as necessary and the latest version of this document will be found on the 4D4Life project wiki at <http://wiki.4d4life.eu/wiki>

Table of contents

1. Introduction	3
2. Requirements	3
2.1. Scope	4
2.2. User stories	6
2.3. Derived requirements	6
2.4. Other requirements	7
3. Design	14
3.1. CoL content	14
3.2. The e-2 platform	15
3.3. The CoL community	18
4. Specification	20
5. References	20
6. Appendix I: User stories	21
6.1. Anonymous user	21
6.2. Named user	23
6.3. Species database	24
6.4. Regional centres (hubs)	24
6.5. CoL editors	27
6.6. CoL operations	27
6.7. Mirror sites	27
6.8. General users	27
7. Appendix II: Specifications	28
7.1. Inventory of components	28
7.2. Data schemas	28
7.3. Service interface definitions	29

1. Introduction

This document is the first edition of Work Package 7's Deliverable 7.1. It represents the results of our research so far into the requirements for the new "e-2" architecture for the Catalogue of Life, and our initial proposals for the specification and design of the new architecture, at present at a high level.

Because of some delays in the start of several work packages, including WP7, and the timing of some deliverables from other work packages which are only becoming available to us at the end of April, we expect to make changes to this document, and to add more details of the design, over the next few months, as we digest the outputs and deliverables which clarify requirements from other work packages.

Furthermore, our next deliverable in WP 7 (D7.2) is a "proof of concept" of the design, consisting of several software components operating together to demonstrate the working of the new architecture design. It is to be expected that the development of this proof of concept will involve some experimentation, may reveal requirements which it might not be possible to meet, and might result in concomitant changes to some of the details of the design. Any such changes will be recorded in future versions of this document.

2. Requirements

(This Section effectively complete in D7.1A, 30 April 2010)

In order to elicit requirements, it has been important to understand the scope of the e-2 architecture, i.e. what the Catalogue of Life is planning to achieve with it. Complementary to this is an understanding of the stake-holders more generally: suppliers of CoL data; users of CoL data and services, etc.

We have assembled lists of requirements from two main sources:

- an initial survey of recent documents and records of meetings in which requirements were expressed, and
- a list of "User Stories" (Section 2.1) which the architecture should support.

From these we have created two lists of requirements. From the user stories, we created a list of "Derived Requirements" (Section 2.3) which are needed for the design of an architecture capable of supporting the user stories. The remaining requirements from the initial survey are listed as "Other Requirements" (Section 2.4). Where overlapping requirements have been identified in the two lists, we removed them from the "other requirements" list, and added to the list of "derived requirements" any additional details gathered from the initial document survey.

Clearly these lists can never be complete, and also it may prove necessary eventually to prioritise fulfilment of some user stories over others. At present, we have derived requirements from these sources of information without prioritisation.

2.1. Scope

The e-2 architecture will provide enhancements to the existing Catalogue of Life system. Key enhancements will be:

- renewing the existing architecture to minimise the load on data providers (typically GSDs – Global Species Databases) by providing a variety of options for data provision and updating, while at the same time improving data currency, and provide better availability and scalability of services, using a distributed architecture,
- placing greater emphasis on the specification and use of automated rules for checking and maintaining the consistency of the Catalogue, to reduce the workload of experts and editors and allow them to concentrate on the difficult cases,
- a more flexible query mechanism providing generic querying over all the attributes stored in the Catalogue, not just names,
- a more fully service-based architecture allowing components to be reused in different configurations and allowing better integration with client systems, and
- the adoption of an ‘open platform’ approach to cater more easily for future needs and interactions with third parties.

The e-2 architecture, as used by the Catalogue of Life, does not:

- insist on real-time access to source data (and not all GSDs may be online), but will allow the Catalogue to stay up-to-date and minimise the number of out-of-date entries,
- impose a single view of taxonomic names or hierarchical positions on species, but it provides a means of presenting and delivering information supplied by expert taxonomists, including the development of tools to monitor quality, identify taxonomic conflicts, support navigation within multiple hierarchies, etc.,
- provide a complete catalogue of all attributes available in the GSDs, but uses a small number of key attributes to allow users to restrict which data sets need to be interrogated further, or
- provide individual occurrence data (but it may provide regions of species distribution).

2.1.1. Relationship to other projects

A number of existing and future international programmes and communities have already recognised the value of the content provided by the Catalogue of Life, and have established or are discussing the creation of linkages with the Catalogue:

- Barcode of Life programme
- EBI (European Bioinformatics Institute) / Genbank
- EDIT (European Distributed Institute of Taxonomy)
- Encyclopedia of Life (www.eol.org)
- GBIF (Global Biodiversity Information Facility, www.gbif.org)
- IUCN (International Union for the Conservation of Nature)
- LifeWatch
- Map Of Life (proposed project)

- TDWG (Biodiversity Information Standards, www.tdwg.org)
- GNA (Global Names Architecture)

The e-2 architecture will enhance the value of the Catalogue to these programmes by providing them with a “taxonomic backbone”, allowing them to import sectors of the CoL and keep their copies up-to-date, allowing them to feed back proposed corrections and new data, etc. Making connections from other programmes to the Catalogue will be more straightforward with the e-2 architecture. In addition to the synchronisation of data, it is intended that the new architecture will provide a platform to make the sharing of software tools and components easier.

2.1.2. Stakeholders

We do not simply refer to “users”, because we wish to include not just end users but also data providers and managers of the CoL, regional centres and related projects who have an interest in using CoL data, tools and systems. The main stakeholders are:

2.1.2.1. Species databases

- global: particular groups of taxa on a global basis (GSDs)
- regional: particular groups of taxa on a regional basis
- national: general species lists on a national basis

2.1.2.2. Information centres, hubs and mirrors

Information centres are managed by organisations in order to manage and deliver data. We use the term “hub” to describe the logical organisation of the data from either a global or regional (or thematic) view. Catalogue of Life information centres are managed by organisations associated with the CoL and typically host portals which give access to the hubs. Hubs may be global, regional or thematic, and provide data sets, user interfaces and services,. Centres may act in one or more of the following roles:

- delivering the global Catalogue of Life
- integrating regional and national species databases
- providing additional data and services (some of which may relate to other roles the organisation has, separate from its role in the Catalogue of Life)
- hosting mirrors of the global CoL and other hubs

2.1.2.3. CoL infrastructure

- management
- operations
 - systems development and maintenance
 - editors

2.1.2.4. Users

Work Package 2 has identified and categorised the end-users of the Catalogue. We have

modified some names to avoid confusion, notably changing Hub to Start Page and Biodiversity to Biologists.

- Integrators: GBIF, EoL
- Linkers: IUCN, NCBI
- Checkers: check spelling, accepted name
- Taxonomists: compare preferred classification with competing ones
- Biologists (Biodiversity): where are most species? Where are endangered species?
- Casual user seeking species information: explore the taxonomic information supplied by the Catalogue of Life interactively, along with rich additional data supplied by hubs and by resources outside the CoL, in order to learn about species of interest
- Other categories: policy/decision-makers, commercial end-users

2.2. User stories

User stories are listed in Appendix I (User Stories), grouped according to the types of users identified in the Stakeholders section above.

2.3. Derived requirements

From the User Stories, we derived a set of requirements to be supported. Here we list the data and service components suggested by the user stories. The numbers in square brackets indicate the supporting user stories listed in Appendix I. We have added some additional details from the requirements survey which were not present in the user stories, to avoid creating duplicated requirement.

2.3.1. Data model

- Edition / publication date [170,171]
- Alternative taxonomic models [150-153,251-259,482]
- Unlimited ranks in the taxonomic hierarchy (both within and above GSDs) [122]
- Additional fields [130]
 - conservation status [145,146]
 - geographic distribution [140,142,143,145,146,471,477,478]
- Annotations [220,320]
 - identified by author [261]
 - text, tags, structured data [230]

2.3.2. Services

- Navigation - starting from an identified node, move up or down the taxonomic tree [122]
- Search - find all nodes beneath an identified node that match a search pattern [100,111], for example
 - wildcard matching of string attributes (name="Vic*", region="5*")
 - range-based matching of numerical attributes (<2000, 250-300)
 - logical combination of attributes (name="Vic*" AND NOT (region="5*" OR region="43 NWG-PN"))
- Import - import structured data into the database [300,301]
- Export - display data from a tree, including aggregate data, formatted using a report template [181,461-464] (including the production of checklists for particular combinations of taxonomic subtree, region, time, etc.)
- Database management - export and synchronise databases [465,810,811] (including the production of CoL "snapshots" at any time to create the Annual Checklist and similar products)
- Directory service - discovery of sites and services [830,910]
- Tree management - copy and modify trees, attach trees, transfer trees [180,251-259,380,530,531]
- Tree comparison - probabilistic matching and visualisation to identify discrepancies [441,442,447]
- Node management - add, remove and move nodes around a tree, annotate a node [220-223,230]
- User management - create and manage users and groups [270,271]
- Account management - save modifications (alternative trees, annotations, tree and node operations) [210]
- Access management - control visibility of user data to individuals, roles, groups and the general public [221,222,260,261,270,271]
- Version management - obtain historic versions of published trees [171,474]
- Status and Usage monitoring - operational support and strategic planning [520-522,610,620]
- Change management - notification of changes to published trees [185,190,191,280,302,303]
- Rule management - record transformations applied to data and ensure transformations are valid [510,522]

2.4. Other requirements

Not all requirements are derived from User Stories. Users are unlikely to specify non-functional attributes of the system, such as security and scalability.

Together with the "Derived Requirements" (Section 2.3), this section constitutes a working

list of requirements for the CoL and 4D4Life software components and data systems which the WP7 e-2 platform architecture design will support. It does not necessarily mean that all these features are currently planned for implementation in the CoL or 4D4Life data or software or will be supported in the near future, but we wish to avoid imposing unnecessary restrictions on future development. Some of these requirements do not require support at the architecture level.

2.4.1. Support for data users

2.4.1.1. More flexible data model

- Separation of taxonomic records and classification hierarchy to allow alternative classifications (from providers, in hubs such as the Global and Regional Hubs, and in outputs)
- More detailed description of synonymy and other name-taxon relationships
- Ability to include or exclude non-primary ranks from searches or navigation (e.g. infra-specific taxa are “first-class” taxa distinct from, but linked to, species)
- Support for common names in the higher-level taxon hierarchy
- Mechanism for recording deletion in order that searches based on invalid names will find a corrected entry

2.4.1.2. Functionality

- Programmatic interface providing direct read-only access to logical database
- Selection of different taxonomic views
- Retrieval of metadata about transformations applied to data
 - transparency of transformations applied by CoL to source data
 - feedback to data providers about errors or inconsistencies in source data
- Subscribe to changes to particular taxa or source databases

2.4.1.3. User access

- Anonymous or registered user access – registered users get additional customisation and annotation capabilities
- Different protocols but consistent interface for (machine-readable) programmatic and (human) user interfaces
- Consistency between AC and DC interfaces where appropriate
- Cross-platform user interfaces (follow standards to support different browsers)
- Alternative interfaces
 - accessibility for users with a disability
 - non-standard devices (e.g. mobile phones)
 - language translations of interface

- selection or ranking of data most relevant to user's language and/or region

2.4.1.4. Performance

- Propagation of updates
 - Rapid (or at least timely) sharing of changes in data with others, including
 - faster updating of the Dynamic Checklist for frequently updated databases
 - making CoL data available to consumers such as EoL for synchronisation
 - action on suggested changes from data providers
- Response time
 - High levels of system availability and MTBF (stability)
 - resilience to faults and external influences
 - System efficiency by minimising
 - network bandwidth
 - resource consumption (processor speed, memory, disc space)
 - human effort involved in data provision, use, management and reporting

2.4.1.5. Participation in community infrastructure initiatives

- Semantic Web, Linked Data
- Biodiversity informatics
 - compatibility and interfacing with related data sources, including indexes to taxonomic and biodiversity informatics publications
 - data standards of TDWG etc.
 - report of the GBIF LGTG, including Linked Data and possible use of HTTP URIs in addition to LSIDs
 - GNA (Global Names Architecture)
 - GBRDS (GBIF's plans for a Global Biodiversity Resources Discovery System)
 - LifeWatch
- Bioinformatics: SRS (EBI)
- Geoinformatics, ecoinformatics etc.

2.4.1.6. Operational requirements

- Designed with usability by the target communities in mind
- Web pages discoverable by potential new users (e.g. by search engines)
- Quality (data available, sufficiently up-to-date, faults discovered and corrected)
- System designed for privacy of sensitive information such as user profiles
- Documentation

2.4.2. Support for data providers

2.4.2.1. Interaction between data users and data providers

- Peer review and quality control systems
- Automatic and manual reporting and feedback procedures
 - some reports may be confidential
 - others may act as stimulants to encourage activity

2.4.2.2. Provision of data

The provision of check-list data sets by data providers and their efficient aggregation and coordination into a consistent Catalogue is a critical part of the CoL activities. In order to make best use of the expertise available and avoid placing undue demands on data providers (typically GSDs), it is important to allow the separation of editing responsibilities, generally organised by taxonomic expertise largely following the taxonomic hierarchy, from the physical locations and support for the data sources, generally organised along institutional lines. Different taxonomic experts and data providing organisations may wish to make their data sets available in different ways. The requirements therefore include:

- Means to receive check-lists which permit efficient transfer and updating, allowing for
 - incomplete and intermittent connectivity (so that harvesting data in real-time is no longer essential),
 - efficient and timely incremental updating of data sets as changes are made (including notification of changes and avoiding the necessity for re-acquisition of the entire data set), and
 - support for existing and legacy methods for data importation, including existing wrappers and files exported by providers,
- Means to allow related check-lists to be aggregated by data providers before submission to the CoL (to allow GSDs to be grouped into clusters managed by suitable organisations, thus reducing the number of data providers with which the CoL has to deal directly)
- Means to allow expert taxonomic and CoL editors to interact directly with the data sets at any stage in the aggregation process, as appropriate to the circumstances, while ensuring the integrity of the Catalogue by the consistent application of quality control procedures
- Means (including schemas, tools) to create structured databases as new GSDs

2.4.3. Support for mirror sites

- Defined standard interfaces and behaviour for CoL services
- Mechanism to support mirroring of CoL data
- Base deployment for provision of CoL services with flexible deployment options (e.g. for deployment using local package management or hosting in locally preferred service containers)
- Mechanism to publish existence of mirror site
- Mechanism for users to discover mirror sites

- Monitoring of mirror sites for availability and currency
- Configurable feedback levels to allow sites to balance monitoring vs security (none, exists, summary, extensive, debug)

2.4.4. Support for regional centres (hubs)

2.4.4.1. Roles of CoL information centres

Centres may host

- One or more global, regional or thematic hubs holding databases and providing services
- One or mirrors of the global and other hubs and their databases and services

Hubs hold

- Local databases
- An integrated “warehouse” including relationships (cross-maps) between databases and between hubs
- User interfaces and services

2.4.4.2. Regional hubs and other alternative views

- Cross-mapping, navigation and simultaneous searching of hubs (global and regional hubs and derivative works such as EoL)
 - for navigation from one to another
 - for searching simultaneously across a range of hubs
 - for creating new data sets
- Work with EoL and others to establish methods to edit and share data

2.4.4.3. Portability

- Platform (operating system, browser) compatibility (system components, user tools)
- Compliance with relevant standards (W3C, TDWG, GBIF, etc.)

2.4.4.4. Synchronisation

- Propagating updates
- Synchronise CoL global and regional hubs and mirrors
- Synchronising mirrors and hubs (hubs can act as mirrors)
- Data in the hubs needs to be comparable between hubs especially for cross-mapping the taxonomies and the species concepts
- Support for synchronisation with other data aggregators e.g. EoL, ITIS

2.4.4.5. Management

- Managed locally and responsive to local needs and opportunities

- Easy deployment of new hubs
 - by copying existing hub software for “standard” hubs
 - by suitable interfaces to non-standard hubs (built using existing local facilities; could be wrapped)
 - by suitable open standards which make it easy to interface to hubs having a non-standard implementation
- Tools for communication (between human)

2.4.5. Support for internal CoL activities and management

- Automation of repetitive tasks, with task-based assignment to humans for expert processing and verification
- Transparent and publishable logging and audit
- Automated configuration management to simplify installation and management over distributed architecture
- The system should have a “business model” which is economically sustainable

2.4.5.1. Statistics

- Usage of the various CoL services
 - Dynamic Checklist web-site, Annual Checklist web-site and CD
 - Web services, LSID resolution, new modes of data access
- Requests made by users
 - sectors most used
 - unfulfilled requests (missing information)
- Data providers
- CoL performance (up-time, response times, etc.)

2.4.5.2. Monitoring

- Transferable and consistent monitoring and basic control available to all service centres
- Availability monitoring of all source databases and hub nodes
- Alerts for unexpected system behaviour
 - response mechanisms and procedures
 - 24-hour rotating responsibility for responding to alerts
- Control and management tools
 - automated procedures
 - manual controls
 - user interfaces for monitoring and control

2.4.5.3. *Internal CoL procedures*

- Design changes to enable unification of Dynamic and Annual Checklist production
 - migration of testing techniques and working practices from Annual Checklist to Dynamic Checklist production
- Backup of critical data and systems, disaster recovery plan
- Rule-based reconfigurable (business) processes available to the CoL staff to allow them to change the flow of processing steps involved in creating the Catalogue (perhaps BPEL might be a solution for both this internal requirement and an external “platform” requirement for such workflows).

2.4.5.4. *Software design requirements*

- Distributed component-based service-oriented architecture
- Testability to ensure correct behaviour
- Scalability to handle projected growth in data, uses, services and users
- Maintainability and related abilities (interoperability, modifiability, supportability, reusable software components)

2.4.6. *Community involvement*

- Architecture and design to be published
 - components to be open source to encourage broader involvement
 - extensibility (adding features and customisations)
 - provision of a “platform” capability to allow 3rd party development of additional services to extend the system, and possible hosting of such services
- Clear roles and benefits for data providers to encourage them to contribute their data (including, for example, peer review and feedback mechanisms)
- Identify relationships with other parties
 - legal and licensing issues
 - safety (fitness for purpose, protection from legal action)

3. Design

(This Section is a preliminary version in D7.1A, to be completed when D7.1B is delivered on 31 August 2010)

3.1. CoL content

3.1.1. Checklist and classification

The Catalogue of Life provides a mapping of names to species (and higher-level taxa), and the information is provided in a tree structure reflecting the traditional hierarchical way of classifying organisms. Because of the key role of the taxonomic hierarchy in organising this information, we frequently use the word “tree” as a concise term to refer to all of the CoL or any subset defined by a higher taxon, consisting of a checklist of species, including synonyms and other CoL data elements, arranged in a single taxonomic hierarchy.

A tree represents a taxonomic model. In the existing architecture, there is a single “consensus” tree which defines a widely-accepted taxonomic model based on expert opinion contained in the Global Species Databases (GSDs). Each GSD is curated by experts in the taxonomic groups represented in the GSD. A proposal for the e-2 architecture is that it should support the representation of alternative taxonomic models, to allow users to work with other models and to compare multiple models.

A node of the tree represents a set of scientific names (or taxon concepts), which refer to the same species according to the taxonomic model represented by the tree. Broadly this equates to a node representing a species (or higher level taxon). However, for the purposes of representing alternative taxonomic models, the former definition is more explicit on the relationship between nodes from multiple trees.

Each node in a tree also has a single accepted name, which is one of the scientific names (or taxon concepts) chosen as the preferred name, again, based on the expert opinion contained in the GSDs. The position of species within the tree matches the accepted name of species such that accepted names of the same genus are grouped together.

3.1.2. Geographical range

In addition to other data attributes described below, which have been identified for inclusion in the Catalogue, a key additional enhancement to the content is provision for the geographical range of each species. This is useful not only for its own sake to provide users with additional information about species, but also for its role in the management of the data. The Catalogue of Life is being extended to a distributed structure in which regional data centres or “hubs” will play a vital role in completion of the Catalogue, improving its data content and communicating with users.

The presence of geographical attributes in the data for species will allow subsets of the main Catalogue to be exported, which are defined by geographical extent as well as by taxonomy, such as a check-list of the insects of the Iberian peninsula. Such subsets can also be used as a basis for the creation of geographically targetted identification aids and other products.

3.1.3. Other attributes

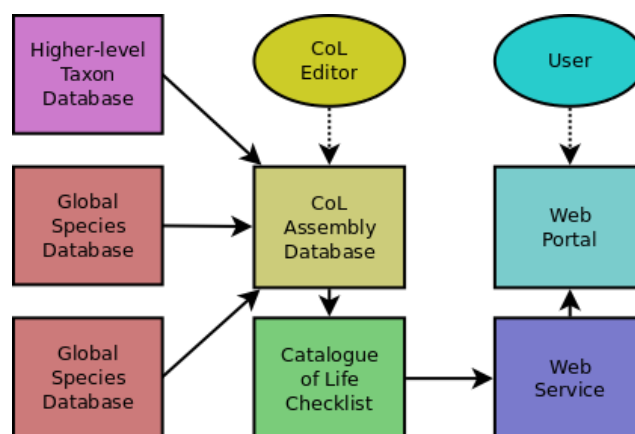
Discussion has taken place in other Work Packages about what attributes should be available in the Catalogue.

At one extreme, replicating all available information from the Species Databases would allow the Catalogue to perform finely-grained queries with fewer false-positives (returned taxa that match the user's specified query but do not match the user's full criteria). However, this would pose serious scalability problems as well as exacerbating the missing information problem from the many heterogeneous systems.

At the other extreme, the Catalogue could just index names. However, this would produce many false-positives and users have expressed interest in narrowing information down by combinations of particular attributes (“all endangered species in Europe”).

3.2. The e-2 platform

The Catalogue of Life is currently developed using a process whereby data is retrieved from the GSDs, through an on-line service (“wrapper”) or an off-line dump. The trees from multiple GSDs are attached to a base hierarchy to create a single tree, which is published as the Catalogue of Life. Users can search the Catalogue using scientific and common names and matching entries from the Catalogue are returned as a list.



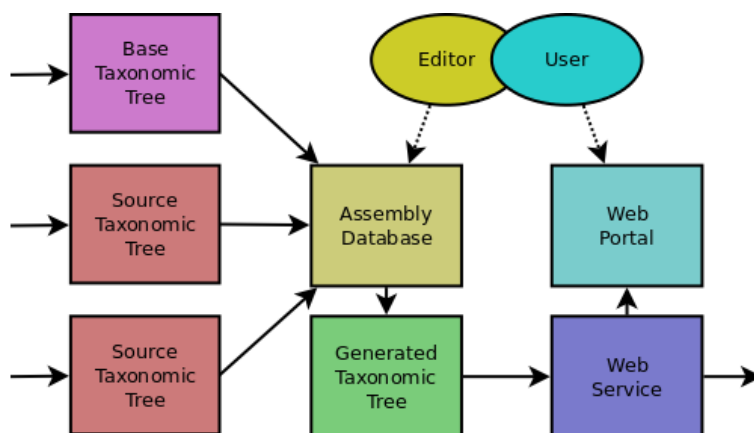
Current model for creating the Catalogue of Life

This architecture suits users looking for a particular single species, to check the accepted name or correct spelling, or to link to the source GSD. The existing web service is limited, and primarily supports the limited interactions available through the user interface. In particular, retrieving large portions of the taxonomic tree is inefficient using the existing interface.

Many users wish to process the search result further, either to deal with advanced queries that return large amounts of data, or to directly process the tree using their own algorithms. In general:

- Users want access to outputs represented using the full CoL tree, or a subset of the tree still represented as a tree.
- Users want to modify tree nodes and links, and attach alternative subtrees, i.e. to perform similar processing to the editing stage.

To this end, the new architecture should expose the data structures of the editing process and make the tools available for general use. The architecture changes from providing a *product* that provides read-only access to data to providing a *platform* that can be used to process data, to generate new data, and to share the generated data.

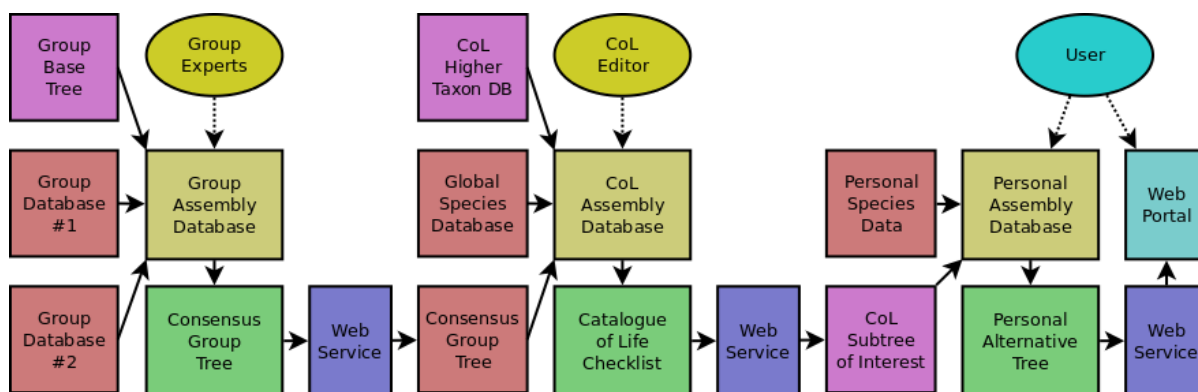


e-2 model as a platform for value-added services

The e-2 architecture, while retaining the basic structure of the existing architecture, emphasises the intention to provide a general platform where users can edit multiple taxonomic trees, in particular merging and attaching trees, in order to create a single tree. This single tree can be searched and navigated as in the existing architecture, but in addition, an enhanced web service allows the release of the generated tree in a form suitable for further processing. This makes it possible to use the output of an instance of this model as input to another instance of the model, allowing the chaining of several processing stages, which may be performed by different users at different sites.

In the e-2 architecture, the Catalogue of Life becomes a single instance of the model, with experienced editors combining a large number of quality input sources to create a consensus hierarchy for general use. Users may choose the Catalogue as a comprehensive base hierarchy on which to build further trees, enhancing it with their own data or building an alternative taxonomic model. For example, a new Global Species Database can be seeded from an existing subtree exported from the Catalogue (as suggested by user story 380).

Alternatively, the e-2 platform may be used without referring to the Catalogue of Life. For example, species databases from two different sites documenting species from related taxonomic groups may use an instance of the model to combine their taxonomic trees into a new checklist curated by experts from both sites. The generated output tree may become an input to the Catalogue.



Example of a system containing repeated instances of the e-2 model

In order to implement the e-2 platform, we intend to use leading-edge, standardised technologies and best practices, including the following:

3.2.1. Service oriented architecture

As shown above, the e-2 Architecture plans to support components that can be reused in many scenarios. To support this, the e-2 architecture is based on a Service Oriented Architecture (SOA). J. Davis provides useful definitions in this context [Dav09].

SOA supports the development of coarse-grained remoteable components defined by their external behaviour, with replaceable implementations. Remoteable components mean that, while following the best practices of abstraction and encapsulation to reduce the amount of coupling between components, they provide a well-defined interface that allows external clients to interact with them over a network. Services provide coarse-grained operations to allow them to scale efficiently when used over a network, where network latency would be more problematic if operations were finer-grained. While the interface is well-defined, the underlying implementation may be replaced, allowing a method for updating to newer technologies and techniques. In addition to the services developed to solve a domain problem, a service oriented architecture typically includes supporting services for providing additional capabilities such as security and discovery services.

3.2.1.1. Service component architecture

A service component architecture (SCA) provides flexible composition of components into higher-level components that reflect the tasks to be performed in the problem domain. SCA provides mechanisms for wiring together primitive services to create more complex services. It also provides a great deal of flexibility to update the actual mechanism for publishing a component as a remoteable service. This allows technologies to be replaced more simply than otherwise expected, for example to replace a SOAP Web Service with a JSON service.

3.2.1.2. Service data objects

SDO provides an open flexible data transfer mechanism supporting disconnected operation and change sets. SDO ensures that the service parameters do not expose the underlying implementation of a service, for example, revealing that the data is coming from a relational database. This obviously helps if the relational database gets replaced by something else. SDO explicitly supports a disconnected programming model, allowing it to be useful even when the services providing data are not always online. This is important in the Catalogue of Life where some GSDs do not provide online access, but prefer to send a “dump” of their database at regular intervals. SDO also supports the representation of change-sets, allowing smaller messages to be sent when only parts of a large data structure have changed.

3.2.2. Business process modelling

Often, the same tasks will be performed in the same sequence over different data sets. Documenting these “business processes” reduces errors in running repetitive tasks and promotes adaptability of processes to new situations.

Business Process Management provides an environment for specifying and executing these sequences of tasks by “piecing together software services one-upon-another to build a new higher-level solution” [Dav09]. BPM provides flexible orchestration of components to implement existing processes, to allow analysis of the existing processes, and to develop

improved processes. BPM tools generally provide graphical user interfaces allowing non-programmers to connect services together to represent a sequence of tasks. The tools encourage the replacement of repetitive error-prone manual tasks with automated processes, but supports situations where a human must make an expert decision or deal with an exceptional case.

3.2.3. Rule management and execution

3.2.3.1. Checking for errors

Much of the correctness of taxonomic trees can be encapsulated in constraints that can be expressed quite simply: “‘order’ is a higher rank than ‘family’”, “all species must contain exactly one accepted name”, “all nodes in a tree must be transitively connected to the top-level of the tree”. Checking the large amounts of data in the Catalogue of Life to ensure satisfaction of all these rules for all entities, in the face of continuing updates, is beyond the capacity of a small group of CoL experts and editors.

However, these rules can be checked by a rule engine in order to identify broken rules that need further attention. When problems outside the existing rule-set are found, additional rules can be created to ensure that these problems are not repeated.

3.2.3.2. Re-applying editorial corrections

Furthermore, making edits to combine trees and make other changes can be manually intensive. When changes in source trees are received from data providers, it is not desirable for the editor user to have to repeat previously applied checks and corrections. Nor is it desirable to reject or delay updates to save the effort of re-applying these edits.

A solution is to encode the edits as rules, which can be re-applied automatically. Editors can be notified of individual data element updates only if a rule fails, thus reducing their workload. Business rules can also be used to restrict editors to prevent them erroneously making edits that do not follow a set of rules (e.g. the codes of nomenclature).

3.2.4. Event notification and message queues

The existing system provides a slow mechanism for retrieving data from large remote databases. However, several techniques are known for providing a more scalable and loosely-coupled model which supports low-bandwidth fast notification of changes and reliable messaging. The application of these technologies can improve the performance of communications between databases, and also support additional capabilities such as informing users when changes occur to parts of the taxonomic tree in which they have an interest.

3.3. The CoL community

The emphasis on reusable components, especially providing editing components for use by external users, provides a model for sustainability of the software. Establishing a community of regular users of the individual components, including those for editing trees, provides a critical mass for sustaining the continued development of the components.

3.3.1. Customisation

The existing architecture provides only for anonymous access with support for limited customisations of results restricted to a single query.

The proposed architecture is being designed with consideration that support can exist for more complex customisations and for personal user accounts as a repository for persistence of these customisations.

Using the SOA model, the customisation framework will be completely orthogonal to the core system. This ensures that the scalability of the core system can be maintained while providing the flexibility of user accounts when required.

The basis of the customisation framework is that users may register for a user account, which provides additional functionality over anonymous access. In particular, it allows users to create personalised views of the data and to save these views for later use.

Apart from user interface personalisations (preferred language), customisations can affect what data is shown. Two examples are annotations and alternative trees.

Annotations allow a user to associate their own data with a node of the taxonomic tree. In its simplest form, users may be able to provide a piece of text which will be displayed the next time they view a species record. This could document that they have previously considered the species record for inclusion in a study. More usefully, these annotations could come from a defined vocabulary, allowing them to be treated as tags and included in search criteria. This would allow users to exclude previously viewed records. More completely, users could even submit pieces of structured data to enhance a record, for example a structure containing a common name and its language, which could be incorporated into the record and found using search.

Alternative trees allow a user to go further than annotations, making modifications to the structure of the tree itself. This allows users to develop their own taxonomic models, for example to investigate the effects of regrouping species under a different set of genera. Again, the results of this customisation should be reflected in the navigation and search services of the architecture.

3.3.2. Sharing

Creating annotations and alternative trees may be useful for an individual, but in some cases it may be useful to share these customisations with others. Examples may include annotations which record errors in the data or alternative trees which represent a widely used taxonomic model.

Making customisations public requires a mechanism to identify these additions as originating from an alternative source, particularly where unmoderated annotations may be mixed with authoritative data supplied by a Species Database. In addition, users should be able to choose whether to see these customisations, or to only see customisations from particular users.

More generally, group management can provide finer control than a simple private/public choice for such settings. Creators of customisations can indicate whether they are private, public, or visible only to one or more users or groups. Similarly, users can indicate whether to see public customisations or only those from one or more users or groups.

A possible enhancement to groups is to include some record-specific roles, such as “supplier”. This role would allow users from the supplying database to annotate records with an indication that it comes from an authoritative source (e.g. confirmations of error fixes). In

the other direction, users may add annotations visible only to members of the supplier group. More broadly, the user and group mechanism could form the basis for a CoL Social Network, providing a forum for users to discuss the Catalogue, the e-2 platform, and shared customisations. This reflects the experiences of the MyExperiment project [GDR07].

4. Specification

(This Section is a preliminary version in D7.1A, to be filled more fully when D7.1B is delivered on 31 August 2010)

The Specification section is included as Appendix II (Specifications). It will be developed continuously during the next phase of the project, and will eventually contain an inventory of components and definitions of the data schemas and service interface definitions, including explanations of the semantics of the data fields, service methods, parameters and events.

5. References

[Dav09] Davis J (2009). Open Source SOA. Manning Publications Co, Greenwich, Connecticut.

[GDR07] Goble C and De Roure D (2007). myExperiment: social networking for workflow-using e-scientists. In: Proceedings of the 2nd workshop on Workflows in support of large-scale science, 25 June 2007, Monterey, California, USA. pp. 1-2.

6. Appendix I: User stories

User stories can be specific examples: “Ted searches for ‘birds’ and obtains the entry for class Aves” or generic: “A user searches for the common name of a higher-level taxon and obtains the entry for that taxon”.

However, we should avoid implementation details: “A user sends a FindSpecies SOAP call...”

Some of the user stories are derived from work conducted by other Work Packages. These are identified using a reference surrounded by square brackets, possibly annotated with section numbers:

- [WP2] - Work Package 2 User services survey
- [WP4] - Work Package 4 Hub survey

User stories derived from the surveys sometimes do not fit well into the format, as they were provided as answers to particular questions. We derive summary user stories to capture the ideas behind such stories, and provide references to the supporting stories in square brackets e.g. [457]. Users stories that are linked in this way have an R suffix on their number, to prevent renumbering errors.

Many of the unreferenced user stories are derived from experience with the existing system and casual comments during meetings.

Key to numbering: second digit: 0 = existing behaviour

6.1. Anonymous user

100 A user obtains a species name published in literature or uttered by a person, and enters the name into the CoL interface. He or she receives a list of species which have been referred to by that name, an indication of the current status of that name in relation to the species (accepted, synonym, vernacular, invalid), as well as other names for the species, and references to authoritative sources.

101 A user searches for records that match a string containing a wildcard. The CoL returns information about taxa that have accepted names, synonyms or common names that match the string.

110 Starting from a specific taxon, a user navigates up or down the taxonomic hierarchy.

111 A user performs a search on the subtree underneath an identified taxon.

120 A user searches for the common name “birds” and obtains the entry for class Aves.

122 A user navigates the tree, selecting whether to include or exclude non-primary taxonomic ranks.

130 A user searches over an attribute other than name, for example geographical regions, conservation status,...

131 A user combines multiple attributes into a single search, using logical expressions such

as (name = "Vic*" AND (region = "5*" OR region = "43 NWG-PN")). An appropriate user interface might mean the user does not need explicitly to have to type in logical expressions.

132 A user searches for all taxonomic families that contain more than 500 species.

140 A user searches for records about a particular species, genus, or higher-level taxon, with an additional constraint of a broad geographic region (TDWG Level 4 or marine equivalent). The CoL returns information about species within the specified taxon that occur within the specified region. The user can represent arbitrary set constraints, including:

- "any taxa that are present in (all three of) Australia, PNG, and Indonesia" – has all of the named regions, may have others.
- "any taxa that are present in (at least one of) Australia, PNG, or Indonesia" – has all or some of the named regions, may have others.
- "any taxa that are not present in Australia, PNG or Indonesia", "any taxa except those that are present in Australia, PNG, or Indonesia" - has none of the named regions, may have others.
- "any taxa that are only present in Australia, PNG, and Indonesia" – has all of the named regions, does not have others.
- "any taxa that are only present in Australia, PNG, or Indonesia" – has all or some of the named regions, does not have others.
- "any taxa not associated with any region"
- "any taxa associated with all regions"
- "any taxa that are not present in at least one of Australia, PNG, and Indonesia" – has some but not all named regions, may have others
- "any taxa except those that are only present in Australia, PNG or Indonesia". – may have named regions, does have others

142 A user uses ISO 3166 country codes to identify a region rather than TDWG Level 4. This simplifies the input of "Indonesia" as "ID" rather than as an inclusive list "42 BOR-KA, 42 JAW, 42 LSI-BA, 42 LSI-LS, 42 MOL, 42 SUL, 42 SUM, 43 NWG-IJ" or as an expression "42 excluding 42 BOR but including 42 BOR-KA, also excluding 42 LSI-ET, 42 MLY, and 42 PHI, also including 43 NWG-IJ"

143 A user searches for a species using a common name, such as "robin" and discovers that there is more than one species with that name. They use the geographical distribution information to distinguish between European and American robins.

145 A user obtains the records for all endangered species that live in a specific region.

146 A user searches for all endangered species in Europe, and obtains the result as a taxonomic hierarchy containing only endangered species present in Europe. The user explores down the Plant branch of the tree.

150 A user selects a published alternative taxonomic hierarchy for search and navigation purposes.

151 A user compares two alternative hierarchies side-by-side.

152 A user views multiple hierarchies as one tree (more precisely, a "digraph", as some nodes have multiple parents), with points of difference highlighted by colours.

153 A visualisation tool allows users to navigate multiple trees, view them from different

perspectives, etc.

170 A user references the Catalogue of Life in a research paper, referring to a particular edition.

171 A user uses a reference listed in a research paper to display the Catalogue as it existed when the paper author viewed it.

180 A user creates a “snapshot” of the entire CoL index at a particular moment.

181 A user creates a checklist describing the birds found in their area.

185 A user exports CoL data for their own use. They modify the data. Realising the CoL has updated, they merge the new CoL data into their exported data.

190 A user “subscribes to” a part of the taxonomic tree, meaning (s)he is notified when any changes occur to records in that part of the tree.

191 A user “subscribes to” a species database, meaning (s)he is notified when any changes occur to records from that species database.

6.2. Named user

210 A user registers with the CoL and is able to customise their preferences which are stored for subsequent logins.

220 A user annotates a record and the annotation is displayed when they return to that record.

221 A user annotates a record and the annotation is displayed to another user when they view that record, with the annotation marked as being created by the first user.

222 A user annotates a record, and selects whether the annotation should be seen only by himself/herself, by a specified group of users, or by anyone.

223 A user reports what (s)he perceives as an error in one or more taxon entries, and this is fed back automatically to the GSD provider. A management tool tracks progress in resolving the query.

230 A user identifies that a common name is missing from an entry and tags the entry with the common name.

251 A user modifies an existing taxonomic hierarchy to create a new alternative hierarchy.

252 A user adds a missing species record to a point in the tree, effectively creating a new alternative hierarchy.

253 A user deletes a species record from the tree, effectively creating a new alternative hierarchy.

255 A user creates a new alternative hierarchy by gradually eliminating nodes that match known false patterns, to avoid eliminating unknown nodes that need further investigation.

256 The Herring Gull (*Larus argentatus*) and American Herring Gull (*L. argentatus smithsonianus*) are recorded as a single species in the CoL. A user creates an alternative hierarchy in which they are separate species (*L. argentatus* and *L. smithsonianus*).

257 The Herring Gull (*Larus argentatus*) and American Herring Gull (*L. smithsonianus*) are recorded as two species in the CoL. A user creates an alternative hierarchy in which they are a single species (*L. argentatus*).

259 A user reverts to a standard published hierarchy.

260 A user selects whether an annotation or alternative hierarchy created by them is visible only to the user, to a specified group of users, or to anyone.

261 A user views a record with annotations from other users, and clearly sees that the annotations are not part of the core record, and can identify the authors of the annotations.

270 A user creates a group and invites others to join the group.

271 A user accepts an invitation to join a group and can see certain artifacts visible to that group, such as annotations and alternative hierarchies.

280 A user registers an interest in a particular node or sub-tree of the taxonomic hierarchy and is notified when the node or sub-tree changes. Notifications may be sent by multiple mechanisms such as email, text message, or instant messaging.

6.3. Species database

300 A site provides a dump of their database in an Excel file for incorporation into the CoL.

301 A site creates a wrapper that transforms their database into a common format for reading by the CoL.

302 A site generates change notifications when a record is created or modified. The notification indicates that the database has changed, but not how it has changed. This is used by the CoL to determine when to retrieve the entire database, reducing unnecessary retrievals, especially for fairly static databases.

303 A site generates change notifications when a record is created or modified. The notification indicates not only that the database has changed, but how it has changed. This is used by the CoL to reduce the need to retrieve the entire database, reducing load on the site.

320 A site retrieves user annotations on data sourced from its database. It tags these annotations privately as “seen” to avoid having to reread them later. One annotation points out a spelling error, which the site fixes in its source database. It publicly tags the annotation to indicate that the problem has been “fixed”.

380 A subtree of the Catalogue that is not currently managed by a Global Species Database is provided to a site in order to create a new GSD.

6.4. Regional centres (hubs)

420 use CoL to fill gaps in a local resource (e.g. for introduced species) [444, 449]

440 use the CoL to assist with data cleaning [WP4 1.1]

441 identify probable matches between a data consumer’s own list and CoL data (via NZOR services) to provide a report on discrepancies back to a consumer. [WP4 1.1]

442 identify probable matches between between all data provider records in order to facilitate the generation of an NZOR consensus view linked to all relevant provider records (c.f. the lexical buckets in the GNI). [WP4 1.1]

443 use CoL with the data cleaning tools developed by CRIA. [WP4 1.1]

- 444R preferentially use the Australian Faunal Directory and Australian Plant Census where these exist, since these are the framework used by Australian government. [WP4 1.1]
- 445 copy the taxonomy for a region to help develop a local index. [WP4 1.2]
- 446 use CoL content to provide a CoL context for non-taxonomic local resources, e.g. biosecurity and biodiversity data management in non-research government operational agencies. [WP4 1.2]
- 447 compare regional taxonomy with CoL. [WP4 1.2]
- 448 create links from species names to CoL reference [WP4 1.2]
- 449R CoL will be a gap filler providing the global context for data we manage regionally for which there is no regional taxonomic expertise or NZOR data provider. Some consumers of NZOR data will require access to data of global extent, e.g. biosecurity agencies, researchers etc. [WP4 1.3]
- 450 Link to common names [WP4 1.4]
- 451 Border security staff need to interpret common names on intercepted goods in order to determine their threat conservation status in New Zealand (e.g. scan packet text on Chinese herbal medicine, OCR and data-match) [WP4 1.4]
- 452 weed distribution modelers try to interpret data on native distribution of weed from regional data linked to common names. [WP4 1.4]
- 453 link to common names linked to language and geographic reference [WP4 1.4]
- 454 Obtain statistical information from CoL [WP4 1.5]
- 455 Gap analysis provides a means of prioritizing taxonomic research/checklist efforts. [WP4 1.5]
- 456 Support up to 200,000 species, 12,000 genera in queries [WP4 2.1,2.2]
- 457 Support for variable updates from real-time, weekly, monthly to annual [WP4 2.4]
- 458 Provide access to all taxonomies from all sources including multiple taxonomic views. [WP4 2.5]
- 460 CoL provides an OAI-PMH based service for 1) updated records, 2) OAI implementation of record sets, 3) OAI implementation of deleted records, 4) OAI implementation of Idempotency of resumption tokens. If not, then at least that the implementation of Spice/CDM supports that these concepts, where CoL providers are also able to support them [WP4 3.1]
- 461 Get the checklist of a taxon: e. g., after input the name of a family or order, we could get a checklist of the family/order in xml format. [WP4 3.1]
- 462 Get all references for a taxon: e. g., input a genus name and get a list of references about the genus. [WP4 3.1]
- 463 Get distribution data for a taxon: e. g., input a genus name and get a list of countries/provinces the genus distributes. [WP4 3.1]
- 464 Get statistics data for a taxon: e. g., input a name of class and get the number of order, family, genus and species in the class. [WP4 3.1]
- 465 Ability to download a complete checklist, group, or whole database. [WP4 3.1]
- 466 GSD gap analysis [WP4 3.1]
- 467 Focus on core data integration, basic web presence and downloadable data for use in

local tools and interfaces [WP4 3.1]

468 Linkage with the specimen information in GBIF [WP4 3.2]

469 Linkage with the DNA data in NCBI [WP4 3.2]

470 Linkage with full-text reference paper in on-line literature databases [WP4 3.2]

471 List of species (at taxa level) occurring in Brazil. [WP4 3.2]

472 Linked data connecting current CoL concepts with past concepts; annual with dynamic concepts; and global with regional concepts [WP4 3.2]

473 Information refers to ranks like subfamily, superfamily, etc. [WP4 3.3]

474 Information about the changes between different version of CoL, e. g. 2009 vs. 2008 annual checklist [WP4 3.3]

475 a more robust synonymy which includes detailed unacceptability reasons for synonymic decisions across all databases. This would allow for services revolving around synonyms. [WP4 3.3]

476 Integrate names with type species images (see LAPI mellon foundation) and protologues (BHL) [WP4 3.3]

477 geographic distribution [WP4 3.3]

478 endemism [WP4 3.3]

479 Links between global and regional concepts [WP4 3.3]

480 Support for multiple languages in interface, but not required at web service level (simplified Chinese, traditional Chinese, English, Spanish, French, Portuguese) [WP4 3.4]

481 Regional hubs provide both global and regional species databases - some regional data may need to be removed in order to avoid conflicts with global species databases. [WP4 4.3]

482 Support for multiple taxonomies from multiple providers, including the concept of an 'active' taxonomy from a provider (e.g. Landcare as a provider to NZOR supplies multiple taxonomic opinions, and we actively use one of them), and also the concept of an NZOR consensus taxonomy, derived from the multiple opinions provided through the network. So NZOR will provide a single taxonomic opinion (to make it simple for most users), linked to multiple opinions from providers, with an indication of which opinions are 'in use' by each provider. The Use Case is that our biosecurity agency want to know about multiple opinions, but they also want to know 'what Landcare thinks'. [WP4 VI]

483 the China Node will provide the web service of: Species data: get full data set of a species by inputting accepted name or synonym; Higher rank data: get the taxon name that a taxon belong to; Lower rank data: get the taxa name(s) belonged to an taxon; Checklist of a family: download the checklist of a family by a family name; Name validation: check if a name is a valid name (accepted name, synonym or common name) in our checklist or not. If not, evaluate the possibility that the name is misprinting or different suffix. [WP4 VII]

484 ITIS can be access through web services documented at http://www.itis.gov/web_service.html. ITIS might be able to customise services for this project. The entire ITIS database is Spice wrapped and available once 4D4Life can test it. [WP4 VII]

6.5. CoL editors

510 An editor reviews a submitted taxonomic tree, and makes a change to a node (e.g. species name) or a link (e.g. parent taxon). The change is recorded in a manner which allows it to be re-applied to future versions of the data and allows external users to see the editorial decisions being applied to the data.

520 A Species Database sends an incorrectly formatted message to the CoL harvesting software. The system sends a notification to the editor, who can investigate the message and work with the sender to fix the problem.

521 A Species Database sends a taxonomic tree that has no attachment point to the Management Hierarchy. The system sends a notification to the editor, who can select one or more attachment points.

522 A Species Database sends a taxonomic tree that is correctly formatted but contains common-sense errors, such as phyla appearing under classes. The system sends a notification to the editor, who can review the tree and decide on appropriate actions.

530 The editor views a newly imported GSD tree and the management hierarchy side-by-side, selecting a node in each, and then creating an attachment point.

531 The editor imports two bat GSDs (A and B), and reviews them side-by-side. The editor decides to attach the Chiroptera order subtree from A. This replaces the existing Chiroptera subtree from the management hierarchy. Seeing that B provides better coverage of fruit bats, he then attaches the Pteropodidae family subtree from B. This replaces the existing Pteropodidae subtree from A.

6.6. CoL operations

610 Operations is informed when a site becomes unavailable.

620 Operations can generate reports on uptime of sites, user connections, etc.

6.7. Mirror sites

810 A site retrieves the CoL database

811 A site synchronises with the CoL database, obtaining changes since last retrieval or synchronisation.

820 A site downloads the CoL service implementation, and deploys it into their service infrastructure.

830 A site publishes that it provides the CoL services.

6.8. General users

910 A user fails to access the CoL due to problems with the site or the intervening network. They are able to discover a mirror of the CoL and perform the same task using it instead.

920 A user writes a program to automate steps they have done using the web interface. Their

program can perform the same steps using a well-defined machine-readable interface.

950 A user searches Google for the scientific name of a species and the CoL web page about the species is returned in the results.

7. Appendix II: Specifications

This section represents a growing repository of documents and definitions which will guide the development of components of the e-2 infrastructure. As such it will be regularly reviewed, updated and enhanced as the specifications are developed, and as they are tested during component and system development.

7.1. Inventory of components

7.2. Data schemas

7.2.1. Data elements

- Additional defined fields and values, as agreed, may include:
 - conservation status
 - marine, freshwater, terrestrial (not mutually exclusive)
 - new standards for data types, e.g. gazetteer (for Europe)
 - legal status
 - date of retrieval from source into Catalogue
 - thumbnail images
 - estimated completeness of data for taxa
- Arbitrary information from sources or (marked accordingly) from users
 - text from sources supplying additional information about records
 - text contributed by users providing additional information or corrections
 - tags contributed by users to enhance searches
 - structured data from users to enhance data, e.g.
 - additional [common name, language, region] entry for a species
 - additional species record
 - user annotations can be private, shared within a group, or public

7.2.2. Descriptions of the semantics of the data fields

7.2.3. Base Schema documents

(provided by Work Package 7)

7.3. Service interface definitions

7.3.1. Definitions of service methods and parameters

(to be developed)