Adventure of Categories*

Modeling the life-cycle of categories during scientific investigation

Prashant Gupta[1, 2], Mark Gahegan[1, 2], Gillian Dobbie[2]

Centre for eResearch[1], Dept. of Computer Science[2], University of Auckland
Auckland, New Zealand
{p.gupta, m.gahegan, g.dobbie}@auckland.ac.nz

Abstract. Categories are the fundamental components of scientific knowledge and are used in every phase of the scientific process. However, they are often in a state of flux, with new observations, discoveries and changes in our conceptual understanding leading to the birth and death of categories, drift in their identities, as well as merging or splitting. Contemporary research tools rarely support such changes in operationalized categories, neglecting the problem of capturing and utilizing the knowledge lurking behind the process of change. This paper presents a tool – AdvoCate – that represents the dynamic nature of categories and allows them to be modelled and to evolve, while maintaining a category versioning system that captures all the different versions of a category along with the process of its evolution; this helps to better understand and communicate different versions of categories and the reasons and decisions behind any changes. We demonstrate the usefulness of AdvoCate using examples of category evolution from a land cover mapping exercise.

Keywords. Category evolution; versioning system; categorical model; process of science; scientific workflow

1 Introduction

The flux of things is one ultimate generalization around which we must weave our philosophical system [2]. (A. N. Whitehead, Process and Reality)

1.1 The Problem

In every domain of science, we use the approach of classifying and labeling things and events – categorization – to make scientific knowledge more tractable and easier to exchange in the community. Shrager and Langley [3] suggest that the formation (and revision) of taxonomies is one of the prime activities of a computational scientific discovery system, and several activities, such as theory and law formation and experimentation rely heavily upon a sound and stable taxonomy. However, conceptual models, such as a taxonomy or legend (a set of categories displayed on a map,

* The tool's name, AdvoCate (Adventure of Categories), is an allusion to Whitehead's 'Adventure of Ideas' [1]

which can also be arranged hierarchically), are often in a state of flux, even when in operational use, fuelled by better data sources, research discoveries, new observations and changes in domain conceptualization. For example, in biology, there is an almost constant reorganization of the tree of life with deeper genomic insights leading to the reorganization of the taxonomy, along with the discovery of new species and technical advancement. Similarly, the categories in land cover databases often change in response to new scientific understanding and social or scientific needs; which if not accommodated would cause dissonance between the map products and the latest domain conceptualization.

This flux in the meaning of categories has serious implications for the scientific enterprise, if not well managed. However, current knowledge artifacts are (technically) rigid and do not adapt to change, continuing to be static and not representing the more fluid, dynamically changing nature of a domain conceptualization. Even in some cases where tools are available to support revisions of taxonomies [4], they only capture those intensional changes that can be described by triples (for example, the addition or deletion of categories or their splitting and merging). Changes causing a drift in a category's meaning, such as change in the category's formative training examples or in the classifier used, are often neglected. Such a situation may cause several issues, including the following:

- If a taxonomy does not keep up with the latest domain conceptualization, it will not only miss relevant information, but also may result in incomplete or incorrect information, since the categories understood by the researchers may be out of sync with those used in the related information retrieval systems.
- If a taxonomy is updated, but the process of change (evolution) is not captured, we may not be able to understand how and why a change happened, and without that knowledge, we may never understand the transition and evolution of scientific knowledge, thus causing a conceptual gap.
- If the process of change along with the previous versions of categories is not captured, we may not be able to understand the intended meaning of categories at the time when they are used – not as they are currently understood. This may cause a problem in understanding or using scientific artifacts or applications that are based on the previous versions of categories. This situation currently causes many problems related to data reuse and longitudinal studies in the geoscience domain [5, 6], and rather ironically leads to much research on ontology harmonization. (We may not need to harmonize categories post-hoc if we capture their evolution process.)

This paper presents a tool – AdvoCate – to support the continuous flux in categories, and records the evolution process along with a category versioning system. Before we proceed further, it would be useful to state the scope of this work. Firstly, this work tries to capture and provide evidence to possible changes in categories while a researcher works at his workstation, which in turn may provide candidate changes for communities facing ontologies and databases. However, the changes made at an individual level goes to local versions and communities have different protocols to revise their knowledge resources. This work does not focus on how and if such changes are updated or revised in a community or organization, where several stakeholders along with organizational and social aspects interact as discussed in [7]. Secondly, in this work, we have considered the hierarchical conceptual models, such as taxonomy and legend, and modeled their evolution. However, we understand that categories exist in other configurations and so are their modifications, which we haven't considered.

1.2 Current Approaches

Knowledge evolution is a commonly known problem and has been investigated in various scientific communities. In machine learning, concept drift refers to a similar problem, where a categorical model[1] that represents the target classes (or categories) becomes inconsistent as the underlying data distributions or the hidden contexts (not explicit in the predictive features) change [9]. Existing solutions in the literature, such as ensemble learning and instance selection [9], support the revision of a categorical model, so that it can correctly classify the new data. However, their focus is only on the revision, rather than understanding and representing the change and evolution of the model. Fanizzi et al. [10] employ a conceptual clustering technique on populated ontologies to detect concept drift and new concept occurrence in a domain with the focus on only the discovery of change. Wang et al. [11] introduce concept drift in the context of the Semantic Web, which covers any kind of change in the meaning of a concept, including change in its intension, extension, label and relationships with other concepts. However, their work focuses on analyzing the change in concepts that already took place and is explicitly represented in some conceptual model; rather than supporting, capturing or utilizing the process of change.

The problem of change and evolution is widely recognized in the database and Semantic Web research communities, which address this problem at the macro level, rather than the micro level analysis of an individual concept. Database schema evolution has been a long-standing research challenge for the information systems community for some time. There exist several commercial and research solutions to schema evolution [12]. One of the most promising research solutions is the PRISM tool [13], which supports schema evolution with a focus on data preservation and supporting legacy queries and updates. It provides a set of change operators to describe schema modifications and also supports schema versioning. In the semantics community, ontology evolution and versioning research has made many advances in the last decade. Several attempts have been made to conceptualize and structure ontology evolution into a process model that describes the various tasks involved, both to provide a complete understanding of various components of ontology evolution and also as a design requirement for software frameworks to support ontology evolution [14]. Some examples of projects that support the process of ontology evolution are KAON [15], Change-management plugin and PROMPT plugin to Protégé [16], OntoView [17] and Evolva plugin to NeON toolkit [18]. KAON supports the whole ontology

[1] Categorical models are created by using training data and estimating the parameters of the classifier used. This helps to classify the test data (or pixels in an image) into different categories. Depending on the classifier used, the categorical model can be a probability distribution model or equations that define partitions in the multispectral space [8].

evolution cycle, from discovering changes to updating ontologies and propagating changes to the dependent artifacts; however, it considers versioning as a separate task and only records the latest conceptualization. Protégé plugins and OntoView support change management for distributed ontologies with a special focus on versioning but do not support the change discovery aspect. Evolva supports ontology evolution with a focus on discovering changes from existing domain data that are external to the ontology. Each of these frameworks has their own underlying process model for ontology evolution, reflecting their perspective and focus.

The main problem with the current solutions (schema and ontology evolution) is that they are largely disconnected from the process of science. Commonly, changes captured and implemented in ontology evolution tools are straightforward, such as addition or deletion of concepts, motivated by ontology learning mechanisms. None of these tools, to our knowledge, connect with the scientific processes to capture and implement changes in domain conceptualization: all of these changes are considered to be top-down and are often left for ontology engineers to deal with. It is understandable that it is hard to capture and digitally represent the process of change – such as how a change is conceptualized and the factors that led to it – if it happens outside the computational realm. However, in several domains of science, changes in categories are conceptualized computationally. For example, the classes in land cover databases often change as researchers interact with training data and classification methods and as their experience with domain concepts deepens. But, usually the tools that might capture such changes in categories (data analysis tools for example) are disconnected from the tools that record and support evolution of ontologies.

1.3 Our Approach

This work tries to bridge the gap between the process and products of science and blend together ontology and workflows within a model for the process of science that supports category evolution. It is inspired by Whitehead's process philosophy [2], which considers science as a perpetual process and scientific artifact as a snapshot of this process. It suggests that the current representation of a taxonomy only captures its temporal understanding (understanding at a specific point of time), and neglect the deeper understanding that lies in the process of its construction and evolution (concerns often heard in our research discussions [19]). This work also recognizes the cognitive dimension of concept modeling that informs how we use concepts in our mind in a highly dynamic and flexible way, such that their meaning changes temporally, spatially and with different situations and contexts [20]. However, we rarely see such aspects in our computational representations, with few exceptions [21].

Our AdvoCate tool, described in what follows, is designed to support the process view and the dynamic nature of categories. It allows category evolution to be modeled while maintaining a category versioning system that captures different versions of a category along with the process of its exploration and evolution; this helps us to better understand and communicate different versions of categories and the reasons and decisions behind any changes. And more importantly, it ties together a temporal series of science products (such as land cover change maps) even though their underlying categories may shift. For the purpose of this paper, we concentrate on modeling categories, though, of course, we acknowledge that other facets of the scientific process, such as hypothesis (for example, as in [22]) and descriptive models could eventually be treated in the same manner.

2 Deepening the Representation of Categories

In this paper, we adopt the following cognitive science view of concepts and categories – concepts are the mental representation of classes of things that connect our past experiences to our present interactions with the world, and categories, on the other hand, represent the classes of objects in the world that concepts describe [23]. We consider categories as instantiations of concepts, represented (albeit incompletely) in some computational system. This work keeps this distinction between concepts and categories in mind while constructing the data model for AdvoCate.

Contemporary science practices often use the three facets – intension, extension and position in a conceptual hierarchy – to represent a category in a computational system at any point of time during its life, and these facets together represent a category's identity (the concept it is grounded in). The intension of a category refers to the set of associated attributes or features (its schema) and the extension of a category refers to all the entities or items that belong to the category based on some rule(s) and adherence to the schema. Commonly, categories are not represented individually in a scientific domain; rather they are woven into the domain's existing conceptual model, where they form relationships with other categories. Even when a category's intension or extension remain unchanged, its place in a conceptual hierarchy may change over time, which implies that our understanding of how a category relates to other categories may change. Even these three facets are not explicitly represented in our current representation schemes (e.g., SKOS, OWL and OBO). We typically have to infer a category's full intension from its relationships with other categories, which is a subjective interpretation and can vary with different users and applications [11].

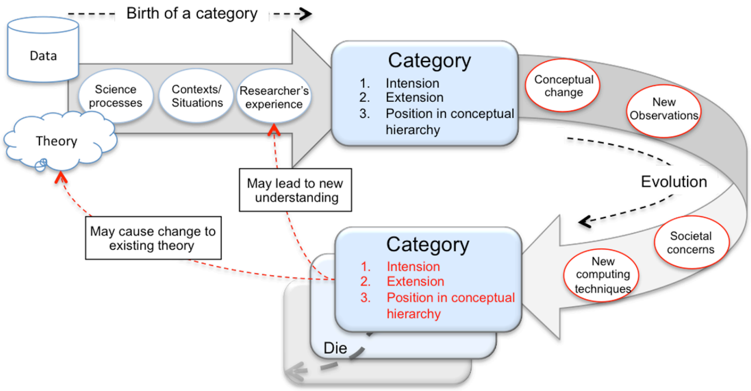


Fig. 1. An overview of the life-cycle of a category, starting form the birth, then evolution and finally the death of the category.

Fig. 1 gives an overview of a category's life cycle, which comprises some key factors responsible for its birth, evolution and death. We argue that the contemporary computational representation of a category does not fully explain the category's existence, nor its identity. The factors responsible for causing changes in a category, the decisions made during the process of a category formation or revision, and the processes (social, physical or computational) themselves play key roles that are important to capture and should be connected explicitly in some way to the category. This information will help researchers understand how and why a category is what it is [24]. It is quite evident that not all of this information is easy to capture and represent computationally as the decision making and reasoning involved in category formation (and revision) is often based on subjective and intuitive considerations, but we may capture some key aspects of the information that will consequently provide a deeper and more complete meaning. For example, a researcher's knowledge, intuition and experience play a significant role in choosing the appropriate computational techniques and methods, which is hard to capture computationally, but we can capture the decisions made by the researcher, which reflect in part the researcher's understanding.

In most cases, knowledge producers do not explicitly connect the factors discussed above with the categories they use for two main reasons: 1) we tend to record this information after a category is created or changed and then accepted by the community, rather than during the process of creation or change. Often, by that time, the process that aided the discovery of some new categorical insight is forgotten. 2) Currently, the systems we use tend to enforce a separation between knowledge representation and analysis activities. The reality is that it is a burden for researchers to record this information during the science process, unless the digital tools they used are designed to automate this recording.

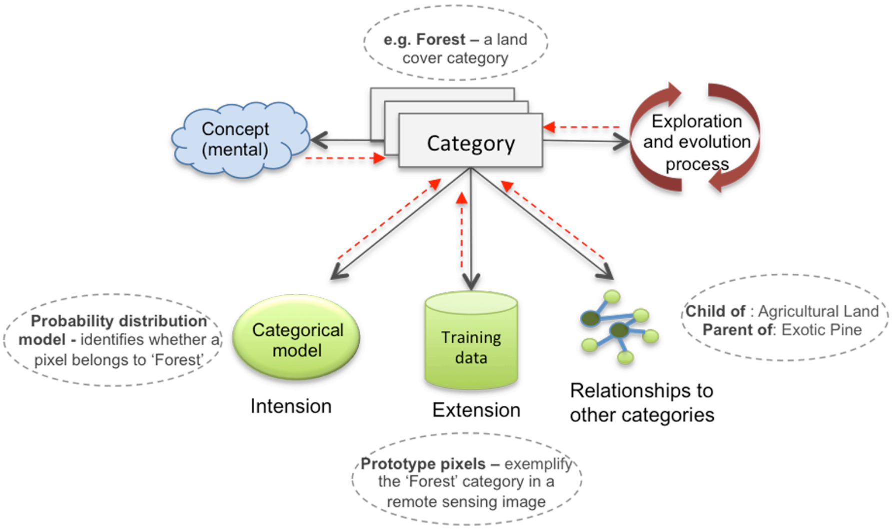

e.g. Forest – a land cover category
Concept (mental)
Category
Exploration and evolution process
Probability distribution model - identifies whether a pixel belongs to 'Forest'
Categorical model
Intension
Training data
Extension
Child of : Agricultural Land
Parent of: Exotic Pine
Relationships to other categories
Prototype pixels – exemplify the 'Forest' category in a remote sensing image

Fig. 2. AdvoCate uses five facets to represent a category. The (red) dashed line indicates that changes to any of the facets result a change to the category's identity, leading to a new version. The ovals show examples of intension, extension and relationships of the 'Forest' category.

Fig. 2 shows the different facets that AdvoCate uses to represent categories. We explicitly represent a category's intension and extension, and its relationships with other categories; changes to any of these facets may result in a new category version. The category is also connected to the (mental) concept it represents, so that we can compare and harmonize categories from different legends representing the same concept. Finally, and most importantly, we connect the category with both its initial exploration and discovery (birth) and its evolution. Exploration refers to the iterative interplay that often takes place between the various classification methods available (classifiers), the training data, other related categories and human concepts as it proceeds towards a stable initial state [25]. The evolution process refers to the changes that occur once the category is in use.

3 Overview of AdvoCate Tool

The fundamental goal of this tool is to move a step forward of the traditional software frameworks used in scientific investigation, which are built to fulfill certain functional requirements, but which are usually not concerned about the knowledge they carry. AdvoCate not only fulfills the functional requirement, modeling and evolution of categories in this case, but also materializes conceptual connections between various knowledge artifacts by explicitly connecting them. Categories are commonly used in databases (in the form of a logical schema) and ontologies (as concepts and properties). The tool also connects categories and their changes with ontologies and databases, via existing tools that support ontology and database evolution. As changes in a category are modeled in AdvoCate, they are analyzed and distributed to the related ontologies and databases; hence, synchronizing the various tools that consume or use categories through the change process. The resulting system thus presents an aggregated view of the scientific process by connecting various scientific artifacts and repositories for data and knowledge through the lens of flux in scientific knowledge. Since, the focus of this paper is mainly on the evolution of categories and legends, we concentrate on describing the services supporting those activities.

3.1 Underlying Technologies

AdvoCate is built using the Python programming language as it provides extensive open-source libraries for scientific programming. For the purpose of building categorical models, we use the Python scientific library, scikit-learn [26], which provides a broad range of simple and efficient tools for machine learning. For UI development, we use Bootstrap (http://getbootstrap.com), a front-end framework for developing responsive web pages, which provides pre-compiled CSS and JavaScript libraries. To build AdvoCate, we used Django (https://www.djangoproject.com), a commonly used Python web framework because it provides several useful components for fast web development, such as an object-relational mapper that supports building data models and provides a dynamic database-access API.

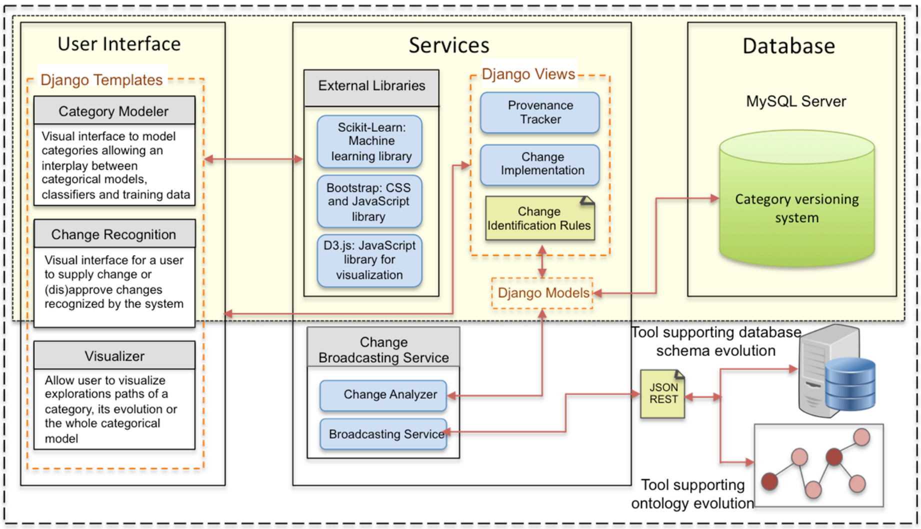

User Interface
Django Templates
Category Modeler
Visual interface to model categories allowing an interplay between categorical models, classifiers and training data
Change Recognition
Visual interface for a user to supply change or (dis)approve changes recognized by the system
Visualizer
Allow user to visualize explorations paths of a category, its evolution or the whole categorical model
Services
External Libraries
Scikit-Learn: Machine learning library
Bootstrap: CSS and JavaScript library
D3.js: JavaScript library for visualization
Django Views
Provenance Tracker
Change Implementation
Change Identification Rules
Django Models
Change Broadcasting Service
Change Analyzer
Broadcasting Service
Database
MySQL Server
Category versioning system
Tool supporting database schema evolution
JSON REST
Tool supporting ontology evolution

Fig. 3. An overview of the three-tiered architecture of AdvoCate, illustrating various UIs and supported services. The outer dashed line subsumes ontology and database tools, portraying that they are conceptually connected to the AdvoCate architecture. The aspects of the architecture with yellow background (enclosed with dotted rectangle) are within the scope of this paper.

3.2 Design

Fig. 3 illustrates the overall architecture of AdvoCate. In this paper, we only discuss details of the sections with yellow background. The rests of the services – Change Broadcasting and Visualization – are still under development and will be discussed in a later paper. The architecture consists of three tiers:

1. User Interface Tier**:** The tool provides three main interfaces (UIs) that support (i) category modeling, (ii) change recognition and implementation, where either change(s) to categories are supplied directly by a user or they are identified by the system (iii) visualization of categories, legends and their evolution. The UIs are built using Django templates and the Bootstrap library. The Django template system provides several useful functionalities, such as tags to include programming constructs and template inheritance to adhere to the DRY (Don't Repeat Yourself) principle. We will discuss the Category Modeling and Change Recognition and Implementation services in the Services section below.

2. Services Tier: The Services tier consists of external libraries – scikit-learn and Bootstrap – to support category modeling and building dynamic user interfaces. The Django view consists of callback functions for various URLs addressing different UIs and describes which data will be sent to those UIs. Our two main services Provenance Tracker and Change Implementation (discussed below in the Services section) are connected with the callback functions in Django views. (A Django view acts as a mediator between the presentation layer and the database

layer by fetching data, which is tied to Django models.) The Django models describe database layout in Python code using the object-relational mapper. The Change Broadcasting service captures changes in categories as they occur and broadcasts them to ontology and database evolution tools in a JSON format using REST services. The service thus connects category models with associated databases and ontologies – through the networks of change and evolution. The outer dashed line in Fig. 3 shows ontology and database tools subsumed within the architecture of AdvoCate to portray that they are conceptually connected.

3. Database Tier: The database tier stores the category versioning system in a MySQL database server. The database stores different versions of categories and their associated legends or taxonomies, along with their exploration and evolution paths. We describe the data model developed for this purpose in subsection 3.4 below.

3.3 Services

AdvoCate provides various services to support modeling category development and change, recording the details of relating to exploration and evolution paths, so that a user may not only look back at the history of categories, but may also examine each step in greater detail – no information about a category is discarded, it is simply versioned. No categories in our system ever become inaccessible, even after their death. In AdvoCate, death is just a state for a category that is no longer subject to evolution, but the category still remains in the system and may be needed to explain old data products it appears in. We now describe the supporting services:

1. Category Modeler: The Category Modeler service provides users the capability to experiment with categories, training data, classifiers and their own understanding of those categories (concepts) and to propose different categorical models, as well as changes in the stable categories. The interface provides access to various classifiers from the scikit-learn machine-learning library, as well as to the associated parameters and validation methods to assess how categorical models change.

2. Change Recognition and Implementation: After categories are analyzed in the Category Modeler interface, AdvoCate identifies changes to categories based on *Change identification rules*. The rules define what to compare to identify changes in categories and how much change quantitatively crosses the threshold to make it worth recording. The rules may change with different types of categorical models and with different usage scenarios. The system currently defines rules only for probability distribution models; adding rules for other models will be added in future work. The system compares the currently modeled state of a category and the latest version of the category stored in the database to recognize changes based on the given rules. For example, the intension (determined by covariance and mean in a statistical distribution) is compared with the intension of the latest version of the category stored in the database. When AdvoCate recognizes the changes to be implemented, it presents them to the user for approval in the Change Recognition in-

terface. The interface allows users to add changes to one or more categories or to the whole legend itself. Often a single change scenario may result in several changes to multiple categories. For example, new training data may cause the birth of a new category, which may in turn change the boundaries of pre-existing categories (drift), which in turn may lead to changes in the intension and extension of multiple categories or the whole legend. To ease the process of defining a change scenario, the tool incorporates a list of elementary and composite change operations, as shown in Table 1. We will describe several change scenarios in the next section as a proof-of-concept.

Change operation	**Syntax and Explanation**
Elementary Changes: Add Category	AddCategory (c) – Creates the new category c and add it to the selected legend
Delete Category	DeleteCategory (c) – Expires the category (It still remains in the system but an expiry date is added to it)
Add Relationship	AddRelationhsip (c1, c2, r) – Adds a new relationship r between c1 and c2
Delete Relationship	DeleteRelationship (c1, c2, r) – Expires the relationship r between c1 and c2
Delete All Relationships	DeleteAllRelationships (c) – Expires all relationships of category c
Change Label	ChangeLabel (c, l) – Create a new version of the category c and change its label to l
Change Intension	ChangeIntension (L, i) – Create a new version of the Legend and change its intension to i. The intension is linked to the legend, rather than to categories (please see explanation in Data Model section)
Composite Changes: Born	Born (c, p) – Add a new category c as a child to category p. AddCategory (c); AddRelationship (c, p, “child-of”)
Die	Die(c) – Delete the category and all its relationships. DeleteCategory (c); DeleteAllRelationships (c)
Merge	Merge (c, c1, c2) – Categories c1 and c2 merge (still remains in the system) to form their parent category c. Born (c); AddRelationship (c, c1, “parent-of”); AddRelationship (c, c2, “parent-of”)
Split	Split (c, c1, c2) – Category c splits into two new child categories c1 and c2 Born (c1, c); Born (c2, c)
Drift	A drift in category can be change in intension, extension, label or its relationships with other categories or a combination of them

Table 1. Elementary and composite change operations incorporated in AdvoCate

3. Provenance Tracker: The Provenance Tracker service, tied within the Django views, tracks and records the process, intermediate results and products, as the categories evolve. This information is modeled as two separate, but related, entities: exploration path and evolution path. At the initial stage of category construction, a category may pass through several cycles of revision before it becomes stable and is added to the domain taxonomy. Exploration paths are the recordings of interplay between various factors involved in proposing categories at this stage. Different exploration paths may provide different categorical models for the same set of input values. Such diversity may be particularly useful in a case where different categorical models have different drivers, such as accuracy or descriptive power. Replaying these exploration paths may help users to choose a categorical model for

their specific needs, or better understand one that has already been selected. Once a categorical model gets stabilized and operationalized (‘published’), changes to it are stored under evolution paths. This may help us to understand how and why a change took place. Uncovering and preserving these details helps us to better understand the implicit meaning in older science products (land cover maps in our case). In addition, this information can be queried to track the effects of a change on a category, which can be referred back to enable the same target effects in a future change scenario. Currently the system only stores this information; an API to query the information will be added in future.

3.4 Data Model

Fig. 4 depicts the portion of the data model that supports the above functionality. Versioning is considered as the first-class citizen in AdvoCate – category, legend, training set, classification model and classifier are all subject to versioning as shown in the schema. The detailed understanding (knowledge-how and -why) of versioning is captured in exploration and evolution paths. The schema explicitly distinguishes concepts and categories, as discussed in section 2. Similar categories (extensions of the same concept) may exist in different legends and can be represented by different intension and extension. Explicit connection between concepts and categories will help to map such categories in different legends and thus help in harmonizing legends. The category table is directly connected to relationships and a training set, which represent (respectively) a category’s relationship with other categories (or its position in a legend) and its extension. The intension of categories, represented by a classification model is connected to the corresponding legend and not to the individual categories. In a categorical model, change in the boundary of a category often results in a drift to boundaries of other categories in the legend. So, if the intension of a category changes, it often means that the whole categorical model is revised and leads to change in the intension of all (or some) categories. For this reason, we linked the classification model to the legend class.

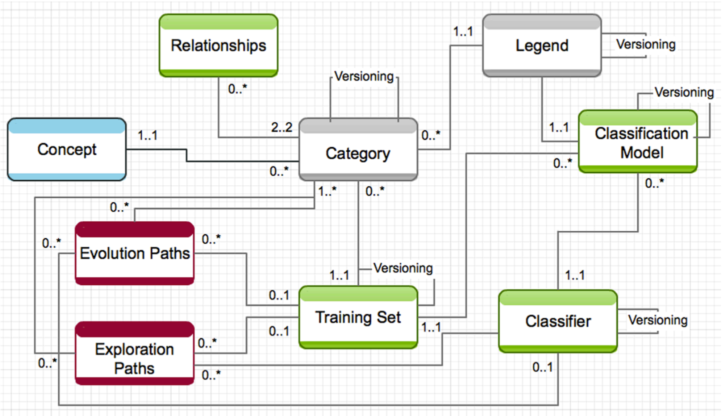


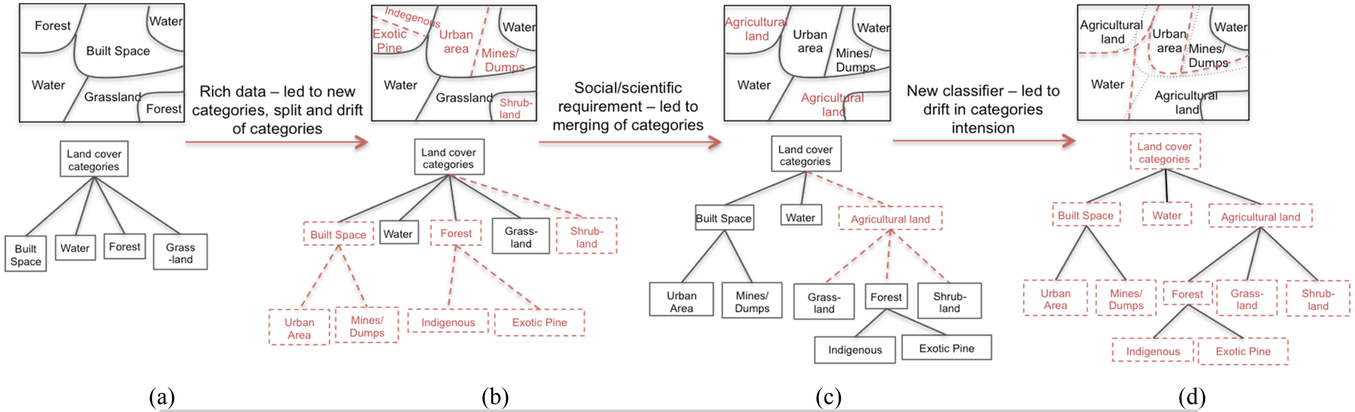
Fig. 4. A section of the data model of AdvoCate tool

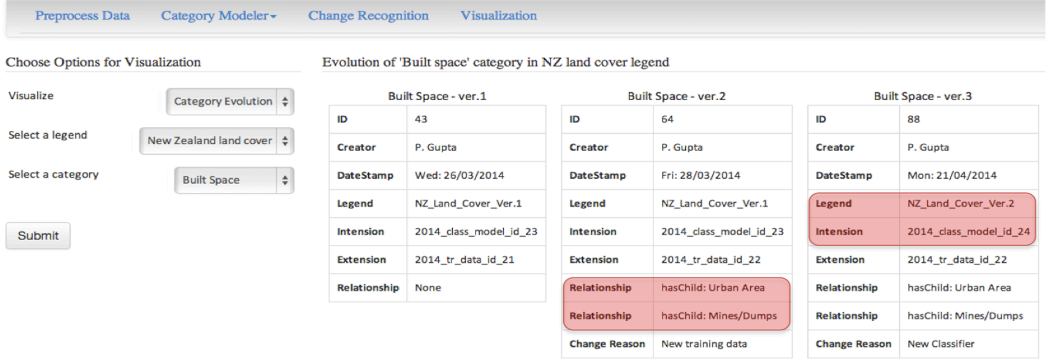
4 Change Scenarios

In this section, we discuss three change scenarios; triggering several changes to the New Zealand (NZ) land cover categories[2], with details of their evolution process. Fig. 5 (a) – (d) show the evolution of land cover categories, describing them through the revisions to maps and taxonomies. The screenshot from AdvoCate in Fig. 5(e) shows how corresponding changes in 'Built Space' (as an example) are recorded in the system as multiple versions of the category along with the details of the changes. Fig. 5(a) shows the initial map, which contains four basic categories: Forest, Built space, Water and Grassland. The initial map was created by classifying remote sensing image data (comprising 4 spectral bands) using the initially proposed categorical models. As new sensing devices with improved spectral characteristics came into existence, the image data was captured in 7 spectral bands. The new rich training data with increased spectral bands (or data attributes) is used for analysis using the AdvoCate system and two key changes emerge: (i) a new category, Shrubland, is born, which was unidentifiable before and was interpreted as Forest; (ii) Forest and Built Space split into new sub-categories. Fig. 5(b) shows that the Forest class splits into Indigenous and Exotic Pine classes, and Built Space splits into Urban Area and Mines/Dumps. The training data with new spectral bands causes the existing clusters (groups of data in a multispectral space corresponding to different categories) to split into multiple smaller clusters, which results into their specializations. Also, we can see the land cover, which was earlier classified as Forest in the bottom right hand corner of Fig. 5(a), is then reclassified to Shrubland, as shown in Fig. 5(b). This shows that the previous bands could not distinguish between these two categories. However the data in new spectral bands can differentiate them, leading to the birth of the new category and extensional drift in Forest. These changes are caused by technical advancements in remote sensing, which provide richer data with better-calibrated spectral bands that improve differentiation of land cover classes. The screenshot in Fig 5(e) shows the resulting new version of the Built Space class in AdvoCate, reflecting the changes in the map and taxonomy.

The next change scenario is to allow government to estimate the land cover under agriculture, so that they can track efficiency of agricultural production. This required merging three land cover classes – Grassland, Forest and Shrubland. The training data for Grassland, Forest and Shrubland, along with their sub-classes, were relabeled as 'Agricultural land'. The new training data was run through data analysis in AdvoCate to update the categorical model. However, this change in the model does not affect the boundaries of other categories. AdvoCate then adds the new category, agricultural

[2] Land cover categories are often modeled as statistical distributions or decision rules by analyzing remote sensing image data that represents spatial distribution of energy reflected from the earth in different spectral bands [8]. Different spectral bands have their own strengths in terms of the information they convey to the remote sensing procedure. For example, in the visible/infrared range, the value reflects the properties such as moisture content, cellular structure of vegetation and the level of sedimentation of water.





Choose Options for Visualization

Visualize: Category Evolution

Select a legend: New Zealand land cover

Select a category: Built Space

Evolution of 'Built space' category in NZ land cover legend

Built Space - ver.1

ID	43
Creator	P. Gupta
DateStamp	Wed: 26/03/2014
Legend	NZ_Land_Cover_Ver.1
Intension	2014_class_model_id_23
Extension	2014_tr_data_id_21
Relationship	None

Built Space - ver.2

ID	64
Creator	P. Gupta
DateStamp	Fri: 28/03/2014
Legend	NZ_Land_Cover_Ver.1
Intension	2014_class_model_id_23
Extension	2014_tr_data_id_22
Relationship	hasChild: Urban Area
Relationship	hasChild: Mines/Dumps
Change Reason	New training data

Built Space - ver.3

ID	88
Creator	P. Gupta
DateStamp	Mon: 21/04/2014
Legend	NZ_Land_Cover_Ver.2
Intension	2014_class_model_id_24
Extension	2014_tr_data_id_22
Relationship	hasChild: Urban Area
Relationship	hasChild: Mines/Dumps
Change Reason	New Classifier

(e)

Fig. 5. (a) – (d) show the evolution of land cover categories, through the revisions to maps and taxonomies. The red (dashed) lines in the maps show new/changed boundaries and red classes are new categories. In taxonomies, red (dashed) edges signify new/changed relationships, and red (dashed) nodes represent new classes or drift in the classes. The screenshot in (e) shows the corresponding changes (highlighted by red transparent rectangles) in different versions of 'Built Space' category, as stored in the AdvoCate system.

land, as the parent class to the three merged categories (with the user's consent) as shown in Fig. 5(c). The motivation in this scenario can be seen as a social/scientific need leading to the merging of three categories.

Lastly, the final change scenario originates from a change in classification method. Previously, a maximum likelihood classifier was used for data analysis, but a new classifier (a non-parametric neural network classifier) was created that provides better accuracy; thus we used it for data analysis on the existing training data. The new classifier changed the boundaries of several classes as shown in Fig. 5(d) and resulted in a more accurate categorical model, leading to change in intension of most of the categories. In this scenario, we created a new version of the legend as well as the categories. The change can also be seen in the resulting new version of Built Space as shown in Fig. 5(e).

5 Conclusion

AdvoCate supports 'deepening the representation' of both categories and the evolution process – an important aspect to support scientific reusability and reproducibility. This information can also be seen as provenance. However, in current practice, we do not capture rich provenance, such as how and why a change occurs. Advocate explicitly captures these details; hence enriching the provenance of change. The unique contribution of this tool is the blend of a cognitive model of categories, ontology (legend in this case) and workflows (process of exploration and evolution of categories) in a single model, bridging the gap between process and products of science. Moreover, its usefulness is demonstrated in the previous section, which shows how AdvoCate connects a temporal series of maps and taxonomies as the underlying land cover categories change, along with the detailed understanding of the changes. In conclusion, AdvoCate not only records the temporal series of land cover maps for a knowledge producer, but also captures the detailed understanding of how the maps are constructed and changed for knowledge consumers to use this information efficiently – improving both knowledge production and consumption.

Future work includes completing the Change Broadcasting service (propagating changes to databases and ontology evolution tools) and visualization of category evolution. This also includes refining the data model through validation with use-cases, adding in change recognition rules for various categorical models, and finally expanding the scope beyond categories to include additional aspects of the science process.

References

1. A. N. Whitehead, *Adventures of Ideas*. Cambridge: Cambridge Univ. Press, 1933.
2. A. N. Whitehead, *Process and Reality: An Essay in Cosmology*. New York: Social Science Book Store, 1929.
3. J. Shrager and P. Langley, *Computational models of scientific discovery and theory formation*. San Mateo, CA: Morgan Kaufmann, 1990.

4. S. C. J. Lam, D. Sleeman, and W. Vasconcelos, “ReTAX+: A Cooperative Taxonomy Revision Tool,” in *Proc. AI-2004 Conf.,* Cambridge, UK, 2004, pp. 64–77.
5. M. Gahegan, W. Smart, S. Masoud-Ansari, and B. Whitehead, “A semantic web map mediation service: interactive redesign and sharing of map legends,” presented at the 1st ACM SIGSPATIAL International Workshop, New York, USA, 2011, pp. 1–8.
6. M. Herold et al., “A joint initiative for harmonization and validation of land cover datasets,” *IEEE Trans. Geosci. Remote Sensing*, vol. 44, no. 7, pp. 1719–1727, Jul. 2006.
7. D. Ribes and G. C. Bowker, “A learning trajectory for ontology building,” *Annual Knowledge and Organizations Conference*, May 2005.
8. J. A. Richards and X. Jia, *Remote Sensing Digital Image Analysis*, 3rd ed. New York: Springer, 1999.
9. A. Tsymbal, “The problem of concept drift: definitions and related work,” Trinity College, Dublin, Ireland, TCD-CS-2004-15, 2004.
10. N. Fanizzi, C. d’Amato, and F. Esposito, “Conceptual clustering and its application to concept drift and novelty detection,” presented at the Proc. 5th European Semantic Web Conf., Spain, 2008, pp. 318–332.
11. S. Wang, S. Schlobach, and M. Klein, “Concept drift and how to identify it,” *Journal of Web Semantics*, vol. 9, pp. 247–265, Sep. 2011.
12. M. Hartung, J. Terwilliger, and E. Rahm, “Recent advances in schema and ontology evolution,” in *Schema Matching and Mapping*, Springer Berlin Heidelberg, 2011, pp. 149–190.
13. C. Curino, H. J. Moon, A. Deutsch, and C. Zaniolo, “Automating the database schema evolution process,” *The VLDB Journal*, vol. 22, pp. 73–98, Feb. 2013.
14. F. Zablith et al., “Ontology evolution: a process-centric survey,” *Knowledge Eng. Review*, pp. 1–31, 2014.
15. L. Stojanovic, “Methods and Tools for Ontology Evolution,” Ph.D. dissertation, Univ. of Karlsruhe, Germany, 2004.
16. N. F. Noy, A. Chugh, W. Liu, and M. A. Musen, “A framework for ontology evolution in collaborative environments,” presented at the Proc. 5th Intl. Conf. The Semantic Web, Athens, GA, USA, 2006, pp. 544–558.
17. M. Klein and N. F. Noy, “A component-based framework for ontology evolution,” presented at the Proc. of the IJCAI-03 Workshop on Ontologies and Distributed Systems.
18. F. Zablith, “Evolva: a comprehensive approach to ontology evolution,” presented at the Proc. 6th European Semantic Web Conf., Crete, Greece, 2009, pp. 944–948.
19. P. N. Edwards et al., “knowledge infrastructures: intelligent frameworks and research challenges,” University of Michigan School of Information, May 2012.
20. E. Rosch and B. B. Lloyd, “Principles of Categorization,” in *Cognition and Categorization*, E. Rosch and B. B. Lloyd, Eds. Hillsdale,NJ: L.Erlbaum Associates, 1978, pp. 27–48.
21. C. Parent, S. Spaccapietra, and E. Zimányi, *Conceptual Modeling for Traditional and Spatio-Temporal Applications*. Springer, 2006.
22. B. Gonçalves and F. Porto, “Research lattices: Towards a scientific hypothesis data model,” presented at the 25th International Conference, New York, New York, USA, 2013.
23. G. L. Murphy, *The Big Book of Concepts*. Cambridge, MA: MIT Press, 2002.
24. M. Gahegan, “Beyond tools: visual support for the entire process of GIScience,” in *Exploring Geovisualization*, no. 4, J. Dykes, A. M. Maceachren, and M. J. Kraak, Eds. Elsevier, 2005, pp. 83–99.
25. X. Dai, “Integrated approach for the exploration of geospatial datasets: The interaction of concepts, methods and data,” Ph.D. dissertation, The PA state Univ., PA, 2007.
26. F. Pedregosa et al., “Scikit-learn: machine learning in Python,” *J. Machine Learning Research*, vol. 12, pp. 2825–2830, Feb. 2011.