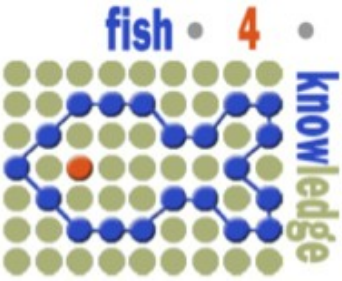

Fish4Knowledge Deliverable D2.2

User Scenarios and Implementation Plan

Principal Authors: E. Beauxis-Aussalet (CWI), L. Hardman (CWI)
Reviewers: J. Chen-Burger (UEDIN)
Dissemination: PU

Abstract: This document specifies the user tasks that will be supported by the Fish4Knowledge user interface. It follows the Deliverable 2.1, that defined the user information needs. This document serves as a functional specification of the Fish4Knowledge user interface. An implementation plan is provided, indicating the research directions for prototyping the user interactions.

Deliverable due: Month 6

Contents

1. Introduction

This document introduces the functional specification of the Fish4Knowledge user interface. It first describes usage scenario, introducing the overall user goals to support (section 2). On this basis, the functional specification of the user interface is given (section 3). It describes the high-level user tasks and user task flow, as well as the underlying data types and data manipulation. These are described independently of the concrete implementation techniques, nor the detailed user interaction design (e.g., without defining the exact layout, buttons, text box and such).

To specify the detailed user interactions, experiments are needed to evaluate user interface paradigms. Our experimentation plan is given in section 4. It sets directions for researching and implementing the user interface.

2. User scenarios

This section describes typical usage of the Fish4knowledge tool, using personas and storytelling.

2.1. Charles – and the study of fish abundance over time

Charles is a PhD student studying fish biology in Taiwanese ecosystems. His research focuses on cyclic evolutions of number of fish over seasons, months and days. He wants to measure community size (i.e., the total number of fish regardless of the species) for different periods of time, and to visualise changes of fish counts over time, using different time units (e.g. fish count for each day or each week). To assert scientific findings, he needs to control and explain his data collection and data analysis processes. He also needs data visualisations to illustrate his findings.

1. Yearly count of fish

Charles wants to measure the *overall abundance* of fish, i.e. the total count of fish taking every species into account. He wants to calculate the overall abundance for each week of the last 3 years.

To do so, he uses the built-in measurement of "overall abundance". He indicates the time unit (i.e., week) and the period of interest (i.e., from January 2010 till December 2012). The Fish4Knowledge tool provides him with a graphic visualisation of the overall abundance.

2. Exploring the counts of fish

Charles wants a more global view, and have a monthly count of fish instead of a weekly count of fish. To do so, he simply changes the time unit of the graphic visualisation, switching from *Week* to *Month*.

Charles is a bit surprised by the overall abundance in 2010. He decides to show

only that year and to switch back to a weekly evaluation of overall abundance. The month of March is particularly surprising. Charles modifies the timeframe of study and the time unit to get a daily evaluation of the overall abundance for the month of March 2010.

3. Controlling the visualisation of overall abundance

Following his exploration of the data visualisation, Charles does not trust the system for using a consistent set of videos. He wants to check what the system has done to evaluate the fish counts.

He starts an explorative view dedicated to the verification of the graphical visualisation. He is provided with a view of the processes that led to the actual data visualisation. He sees global information about:

A) What videos were taken into account (i.e. all videos available for the month of March 2010)

B) What software components analysed the videos and produced the recognition of fish (i.e., the Fish Detection and Fish Tracking components)

C) What computer vision data were used for the visualisation (e.g., the fish that were automatically recognised in the videos)

D) What domain-specific metrics were calculated (i.e., the overall abundance), which formulas were used, how data were used.

E) How data are represented in the visualisation, e.g., the type of diagram, how data are plotted in the diagram.

4. Controlling the videos to take into account

Charles wants to check if a homogeneous set of cameras and videos were available at the beginning of 2010. To do so, he asks for more details about the videos that were taken into account.

While inspecting the set of videos, Charles particularly looks for inconsistencies in the number of cameras or videos available in each location, e.g., if a significant number of cameras were in maintenance during a specific period, if a significant number of videos are missing or are really bad quality.

Charles could select a custom set of cameras or videos to get a homogeneous set of videos, but the set is already consistent enough. The quality of videos is also consistent, but Charles wants to verify it anyway. Thus he inspects a few videos in 2010, and he even looks for video in 2011 to verify if the quality is similar, and if fishes are accurately identified.

5. Controlling the video analysis components version

Charles still wants to find out what makes the overall abundance different in 2010. He has checked that it is not due to inconsistencies in the set of videos, and he continues by checking the software components that analysed the

videos.

He particularly looks for differences due to the version of the components. While inspecting the software components, he notices that the Fish Detection component, responsible for detecting fish among other objects, had a major version change in September 2010. To check how this influenced the result, he launches the analysis of the video before September 2010 with the new version of the Fish Detection component.

When the video analysis is redone, Charles inspects again the visualisation. He sees that the differences between the overall abundance in 2010 and the other years remain the same.

6. Controlling selected data set

Charles now wants to exactly check the data that are retrieved from the database. He is not surprised to see that there is a table named "Fish" that contains one entry per recognised fish, and that these are the data taken into account. But he notices the column called "Detection certainty" which contains a score that represents the probability of error for each fish detection. This score is used as a threshold to select the entries of the "Fish" table to take into account.

Charles decides to set a higher threshold, i.e. to lower the error probability, to see how it influences the results. When he visualises again the overall abundance diagram, he sees that the overall abundance has lower values, but there are no major changes in the trends, and the results for 2010 are still different from the other years.

Charles also tries a lower certainty threshold, but there are still no major changes in the trends, and the results for 2010 are still different from the other years.

7. Studying overall abundance per area

After checking the overall abundance formula and data used, and checking the way overall abundance is represented in the diagram, Charles did not find any technical reason to explain the differences that appear for the year 2010. But now he knows exactly what is represented in the visualisation, and he now trusts the system.

Thus Charles starts thinking about biological reasons that influenced the population dynamics in 2010. He wonders if the changes in 2010 occurred for the whole island of Taiwan, or if there were localised in a specific area. He investigates this hypothesis by calculating the overall abundance for various areas. When he visualises the results, he clearly finds out that the changes occurred in the eastern part of Taiwan, and that differences in that area influenced the national overall abundance he visualised before.

8. Comparing overall abundances for a specific area

Charles wants to compare the overall abundance in the eastern area in 2010,

with the average overall abundance in this area during 2011 and 2012. To do so, he uses the interface provided to check the calculation of domain-specific metrics, i.e., the calculation of overall abundance. He adds the evaluation of the average overall abundance during 2011 and 2012 by defining a custom data set to use.

He then customises the visualisation to get the representation that suits him most to compare the *overall abundance* in 2010 and the *average overall abundance* in 2011-2012.

2.2. Erica – and the study of environmental conditions

Erica is a postdoc studying the interactions between environmental conditions and marine fish communities. Her research focuses on the influence of environmental conditions on fish abundance, fish migration and fish reproduction. She needs to compare measurements of a fish community (e.g., overall abundance, number of species) collected during various environmental conditions (temperature, pollution). Her colleague Charles has asked for her advice about the variations of fish community size (e.g., the overall abundance) during 2010 in the eastern coast of Taiwan.

1. Explore environmental conditions

After visualising the differences of community size in the eastern area for 2010 compared to years 2011 and 2012, Erica starts to look at weather conditions that could have influenced population dynamics. She is provided with weather data extracted from the Central Weather Bureau of Taiwan, and from the underwater sensor network owned by her research laboratory. She notices peaks of temperature and peaks of current velocity that might have influenced the population dynamics.

Erica also searches for other pollution that might have occurred in this area. After searching the web, she finds that a harbour has been renewed and that a chemical leak occurred in the area.

2. Define events to study

Erica wants to study the impact of the unusual environmental conditions during the year 2010. To do so, she uses the functionalities that allow to target and define events to study. She creates an instance of an event for each temperature and pressure peak, one event for the harbour reconstruction, and one event for the chemical leak. For each event, she indicates the period and location of occurrence, and the type of event.

3. Define species to study

Depending on the type of environmental conditions, some species might be particularly impacted. These species are called *indicative species*, as their study highlights the impact of specific events. To explore the species of the fish community, Erica visualises the various species of the community, and their

relative abundance. Among the species of the community, Erica chooses the sets of indicative species that are representative for the environmental events she wants study.

4. Explore the impact of environmental events

Knowing the species and events to study, Erica requests the calculation of various metrics, evaluated around the period and location of environmental events. She creates, compares and saves several visualisations for each event she wants to study.

Once she has studied the visualisations for each event, Erica excludes the hypothesis of the chemical leak, and of some temperature peaks. But she is still not sure about the causes of the variation of community size. To compare the different hypotheses left, she specifies a personalised visualisation to compare custom metrics of her interests.

5. Illustrate findings using a custom visualisation

She finally finds out that three events that occurred in a row might explain the results: the reconstruction of the harbour modified the habitats of several species, and might have impacted other species that feed on them, and two temperature peaks that occurred around the reconstruction of the harbour had amplified that phenomenon. These events occurred during a reproduction period, which explains the variations of the overall abundance. Later on, the fish community size has the same trends as usual, but the species composition has changed.

To illustrate her findings, Erica creates a specifically customised visualisation and sends it to her colleague Charles.

3. Functional specification

This section specifies the user interface functionalities at a high-level. It defines the user tasks, and the related high-level data manipulated throughout these tasks. These functional specifications are given independently of the concrete user interface layout or implementation.

3.1. Methodology

The specification of the user interface follow T. Munzner's model[1], and consists of 4 steps of top-down definition:

1. **Domain problem**: this step describes high-level problems and derives the goals users want to achieve, and the related user requirements. Goals and requirements are defined independently of technical details about the tool that addresses the goals.

[1]T. Munzner, A nested process model for information visualisation design and validation. In IEEE Transactions on Visualization and Computer Graphics, Volume 15 Issue 6, 2009.

2. **Domain-oriented data and operations**: this step concerns means for achieving user goals. It specifies data that can convey user information needs, basic operations needed to manipulate data, and courses of operations for achieving user goals. At this step, user tasks can be expressed as processes using operations and data types.
3. **Visual encoding**: this step defines means to represent data and to execute operations through the user interface. It specifies the concrete user interactions our tool supplies. At this step, mockups can be drawn to define the layout, the user interface components (widgets), and the graphical guidelines.
4. **Algorithm**: this step concerns the implementation of the user interface. It defines the technical specification: technologies and languages to use, and modelling of the program to code (e.g., UML, pseudo-code).

The next sections address the first two steps, and successively describe the domain problems, the user-oriented data, the basic operations on data, and the typical flows of operations. This defines the basic specifications that will allow the independent and simultaneous development of the video analysis components, the workflow and the user interface system. Further technical details are compiled in section 5.

The user-oriented data and the basic operations are subject to modifications throughout the implementation of the project, and following the feedback collected from users. However, we assume that we identify here the main data and operations. This will serve as a basis for collecting feedback, evaluating user interface techniques, and proceed to iterative refinements.

3.2. User requirements

This section briefly summarizes user requirements at a high-level. Attention is drawn on users intent and issues when using the Fish4Knowledge system, without focusing on how users would proceed when using the system. More details regarding domain problems and user needs can be found in the Deliverable 2.1 – *User information needs*.

a) Analyse population dynamics (requirement R1)

Users seek to evaluate several biological measurements. These allow them to establish and analyse facts. For instance they need to evaluate the number of species living in the same area. Users are interested in measurements that concern the analysis of population dynamics and demographical facts. We identified 13 biological metrics of interest. They mainly serve to count fish and species that live in monitored areas, to count the occurrences of various fish behaviours, and to compare these counts over time. The measurements are described in the Deliverable 2.1 – *User information needs (*see section 5.1.a for an overview, section 4 for more details, and Appendix VI for an application to

the “20 questions”).

b) Analyse impacts of events (requirement R2)

Users seek to evaluate the impact of environmental events on fish populations. For instance, they seek to evaluate how a typhoon changes the distribution of species living in the impacted area. The metrics supplied for analysing population dynamics allow to measure the impacts of environmental events. Users need to evaluate these metrics for periods and locations related to events of interest (e.g., before, during and after a typhoon). They also need to record and describe the events of interests, especially their timeframe and location of occurrence. Finally, users seek to consult the available meteorological data, such as water temperature or pressure.

c) Verify analysis (requirement R3)

Users are confronted with trust issues because i) they need to assess scientific evidences, and because ii) the Fish4Knowledge system introduces new technologies they did not trial. These trust issues can be addressed by allowing users to check the underlying computational processes that produced the information they use. To assess the validity of the data analysis, users seek to know how information was processed, by which components, with which certainty, and how computational parameters and video analysis features impact the results they use. The whole chain of computations is concerned by this verification need. And users even seek to verify and watch the video clips themselves.

In addition, users need to assess the replicability of the underlying computations, eventually using different versions of programs or different data sets.

d) Adjust analysis (requirement R4)

Users are proceeding to ongoing data analysis as they monitor fish populations over long period of time. Considering the middle and long-term perspective, users seek for evolutive data analysis that can be modified along with the emergence of new insights, hypotheses or centres of interest. When adjusting data analyses, users might consider biology-specific purposes (e.g., detailing a species composition) as well as technical purposes (e.g., test the influence of computational parameters). Thus, users seek for controlling technical parameters specific to computer vision, as well as domain-specific parameters specific to the evaluation of biological facts.

3.3. Overview of the domain-oriented data manipulations

This section briefly describes the types of data that users will be able to manipulate. It also briefly describes the main data manipulations for achieving

each user goal. Five examples of data manipulations are given:

- Creating a new view by using a template
- Creating a new view by using custom diagrams
- Handling views that require extra computations
- Verifying an existing view
- Modifying an existing view

a) Approach

The domain-oriented data are designed to encode the user information needs into computational data. The domain-oriented data comprise the data produced by video analysis and any other needed data such as environmental data, or user-defined data.

The domain-oriented data are designed to fit users' mental models, rather than video analysis computations. The design rationale is i) addressing the user information needs we identified, while ii) allowing a flexible manipulation and usage of the information and iii) facilitating further refinements of the user information design. The design of the domain-oriented data is grounded in the Deliverable 2.1 - *User informations needs*, specifically in its Appendix VI – *Basic user tasks and functionalities*.

The specification of the domain-oriented data defines data at a conceptual level: it is a conceptual model. By defining data at a conceptual level, we abstract from technical constraints that do not influence user goals, and we focus on users comprehension and manipulation of data.

We follow the notation used for defining *conceptual data model* using *data structure diagrams*[1]*: r*ectangles represent data types (or entity classes in [1]) and the arrows represent data properties (or entity sets in [1]). This notation can easily be translated into UML notation for class diagram: rectangles represent classes, and arrows represent associations (but the triangle shape of arrowheads must be replaced by the one for directional associations).

b) Overview of the domain-oriented data types

On the basis of the Deliverable 2.1, the above scenario (section 2) and user goals (section 3.2), we identified an initial set of 16 data types that allow to encode the user information needs into usable data. Seven data types concern the core biology-oriented data analyses (e.g., goals G1-2 – Analyse population dynamics or impacts of events). Five data types mainly concern the visualisation functionalities (e.g., goals G4 – Adjust analysis), and four data types mainly concern the technical verifications (e.g., goal G3 – Verify analysis). Figure 1 gives an illustration of this data space, as it will be perceived by users. It represents the data objects and their object-to-object properties (i.e. a *Fish* has a *Behaviour*, a *View* has *Diagrams* to display, a *Diagram* has a *Data Set* to represent). A short definition of each data type is

[1] C. W. Bachman, Data structure diagrams. In Data Base, pages 4-10, 1969.

given thereafter. A detailed specification of data objects is given in section 4.

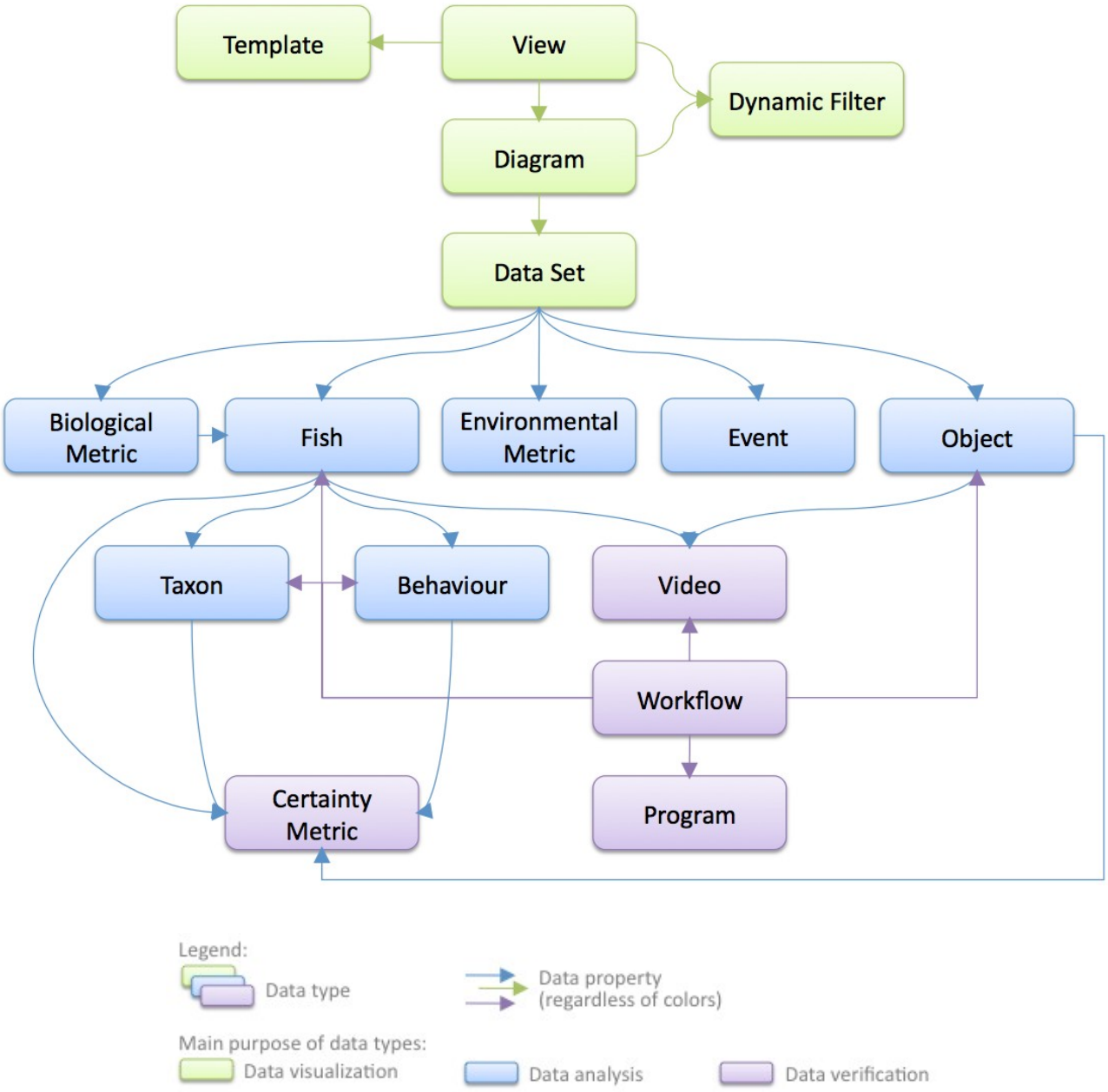

Figure 1 – Main domain-oriented data types and their relationships

View
A *View* is a visualization of *Data Sets* displayed using interactive graphical representations (i.e., *Diagrams*).

Diagram
A *Diagram* is an interactive graphical representation of a *Data Set*.

Data Set
A *Data Set* is a collection of data chosen among any available data types. Typical *Data Sets* contain data such as *Biological Metrics* (e.g., counts of species), *Fish* detected in videos, *Environmental Metrics* (e.g., water temperature) or *Events*. Specific technical *Views* concern *Data Sets* containing *Videos*, *Programs* and *Workflows*.

Dynamic Filter
A *Dynamic Filter* is a specification of the data displayed in *Diagrams*. A *Dynamic Filter* allows users to refine the data displayed in one *Diagram* (i.e., a *Diagram's Filter*) or in every *Diagram* of a *View* (i.e., a *View's Filter*). For instance, while inspecting a *View*, users can interactively redefine the time unit or the timeframe of interest.

Template
A *Template* is a model of a *View* that comprises predefined parameters, and user-defined parameters. A *Template* allows to create a standard, frequently used *View* by indicating only a few user-defined parameters. For instance, *Templates* allow to quickly create elaborated Views displaying several Data Sets and Biological Metrics sharing the same parameters.

Biological Metric
A *Biological Metric* is a measurement used by biologists to study fish populations (e.g., abundance, growth rate). The *Biological Metrics* are those defined in the Deliverable 2.1, particularly in its Appendix VI.

Environmental Metric
An *Environmental Metric* is a measurement used to monitor meteorological conditions (e.g., temperature, pressure). Sensors or meteorological agencies can supply regular measurements of environmental parameters. These data can be visualized and summarized using larger time units (e.g., a daily average of the temperature measured every 30 minutes).

Event
An *Event* is an incident that has an ecological importance. It allows users to define a period and location of interest to be considered for specific data analysis.

Fish
A *Fish* is an entity that was filmed and recognized as a fish by automated video analysis. A Fish is described by four main properties: the taxonomic classification (i.e., species, genus, family) called ***Taxon***, the observed ***Behaviour***, the *Workflow* that analysed the video and described the *Fish*, and the *Certainty Metrics* for this video analysis.

Object
An *Object* is an entity that was filmed, and that was recognised as a non-fish object by automated video analysis.

Program
A *Program* is a software component used to automatically analyse the videos.

Certainty Metric
A *Certainty Metric* is a score representing the accuracy of a video analysis feature recognition. It is evaluated through the video analysis *Workflow that* recognised a *Fish* entity, an *Object*, a *Behaviour,* or a *Taxon* of a *Fish*.

Workflow
A *Workflow* references the sequence of *Programs* that recognised a set of *Fish* or *Objects*. Users can verify which *Programs* produced any *Fish* or *Object*.

Video
A *Video* is a 10 minute video clip filmed by underwater cameras, and cut into a 10 minute excerpt for storage and video analysis matters.

c) Creating a View from a Template

The Figure 2-a below illustrates the operations done for creating a new *View* by using a *Template*. It illustrates a visualisation of *Biological Metrics* and *Events*, but *Templates* can be used to visualise any kind of data. Only three steps are required:

1 – Select Template: A user chooses a type of *Template* from a list of those they can use.

2 – Set Parameters: A user defines the parameters needed to select the *Data Set* to visualise. For instance, parameters could be the timeframe and location of interest, the time unit for the *Biological Metrics*.

3 – Launch View: Once all parameters are set, the user can visualise the *Data Set* through the display features (i.e., *Data Sets*, *Diagrams* and *Dynamic Filters*) that are predefined by the *Template*.

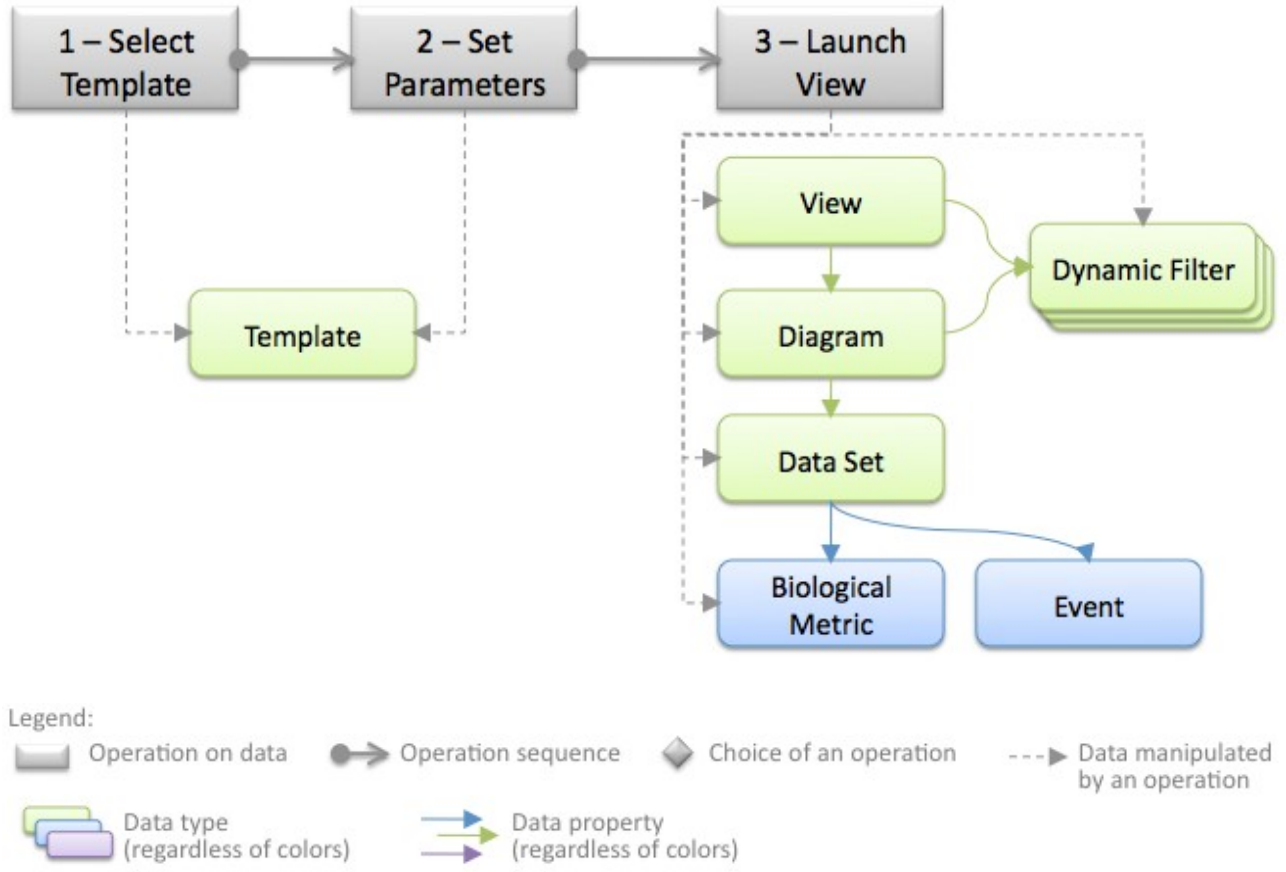

Figure 2-a – Typical data manipulations for using a data analysis template (Requirements R1, R2)

The Figure 2-b below gives an example of the usage of a template that illustrates our second user scenario, step 4 (see section 2.2). For this example, the *Template* is a default visualisation for analysing the impact of an event. Users define 2 main parameters: the *Event* to study, and the timeframes of interest before and after the Event. Eventually, users can define species or behaviours of particular interest because they would particularly be impacted by the event. To do so, they define typical *Fish* to analyse, with their *Taxon* and *Behaviours* (i.e., entity called "*Indicative Pattern*" in Figure 2-b).

On the basis of these parameters, the launched *View* contains a *Diagram* that gives an overview of the number of species and the number of fish from each species, and their evolution over time (i.e., based on the metrics called *Species Composition* – see the Appendix VI of Del. 2.1 for more details). The *View* also contains a *Diagram* giving the number of *Fish* from the indicative species defined in the parameters, and a *Diagram* giving statistics about the occurrences of indicative behaviours (i.e., based on the metrics called *Growth Rate* and *Behaviour Occurrences* – see the Appendix VI of Del. 2.1 for more details). Finally, through a *Dynamic Filter*, users can redefine the time units and the timeframes of all the *Diagrams* of the *View.*

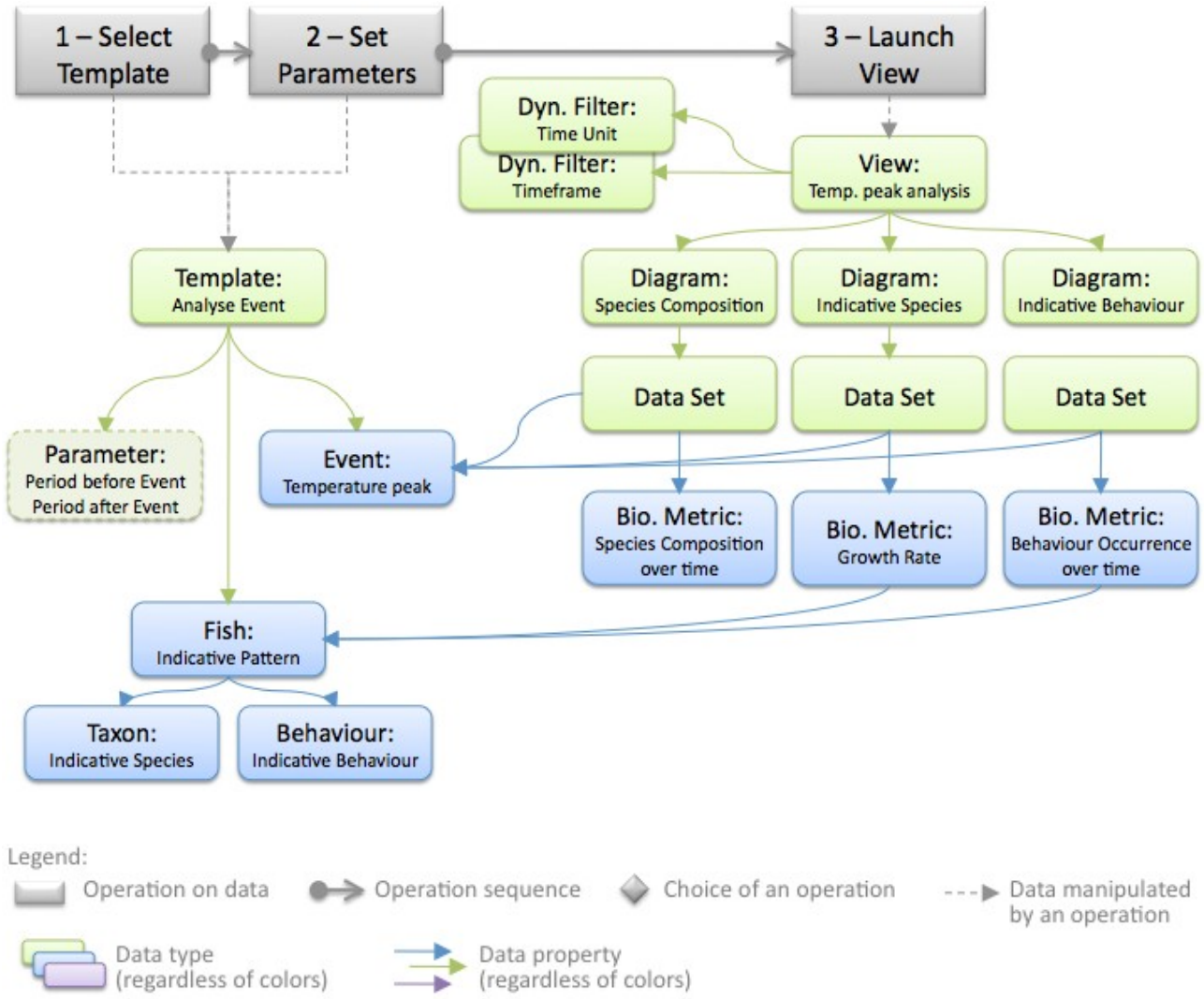

Figure 2-b – Example of data manipulations for using a data analysis template

d) Creating a custom View

The Figure 3-a below illustrates the operations done for creating a new View using no *Template* but custom *Data Sets* and *Diagrams.* Three main steps are required, possibly with a loop between the first two steps:

1 – Select Data Set: A user chooses the set of data to visualise, among any of the available data. Figure 3 illustrates a Data Set containing data extracted from videos. In case of technical data analysis, *Data Sets* could also concern *Videos*, *Workflows* or *Programs* (e.g., to visualise statistics about the *Videos* analysed by a *Program*).

2 – Set Diagram: A user defines the type of *Diagram* to be used for displaying the *Data Set* previously defined. Eventually, display parameters can also be defined (e.g., *Dynamic Filters*, colour sets, size, position).

After completing these two steps, a *View* containing the *Data Set* and *Diagram* can be launched. Users can choose to launch the *View* (next step) or to add another *Diagram* (loop back to step 1).

3 – Launch View: The user can visualise the *Diagrams* that were previously defined. Eventually, the user can define *Dynamic Filters* applying to all *Diagrams* of the *View*. The user can also rearrange the layout of the *Diagrams* (size and position).

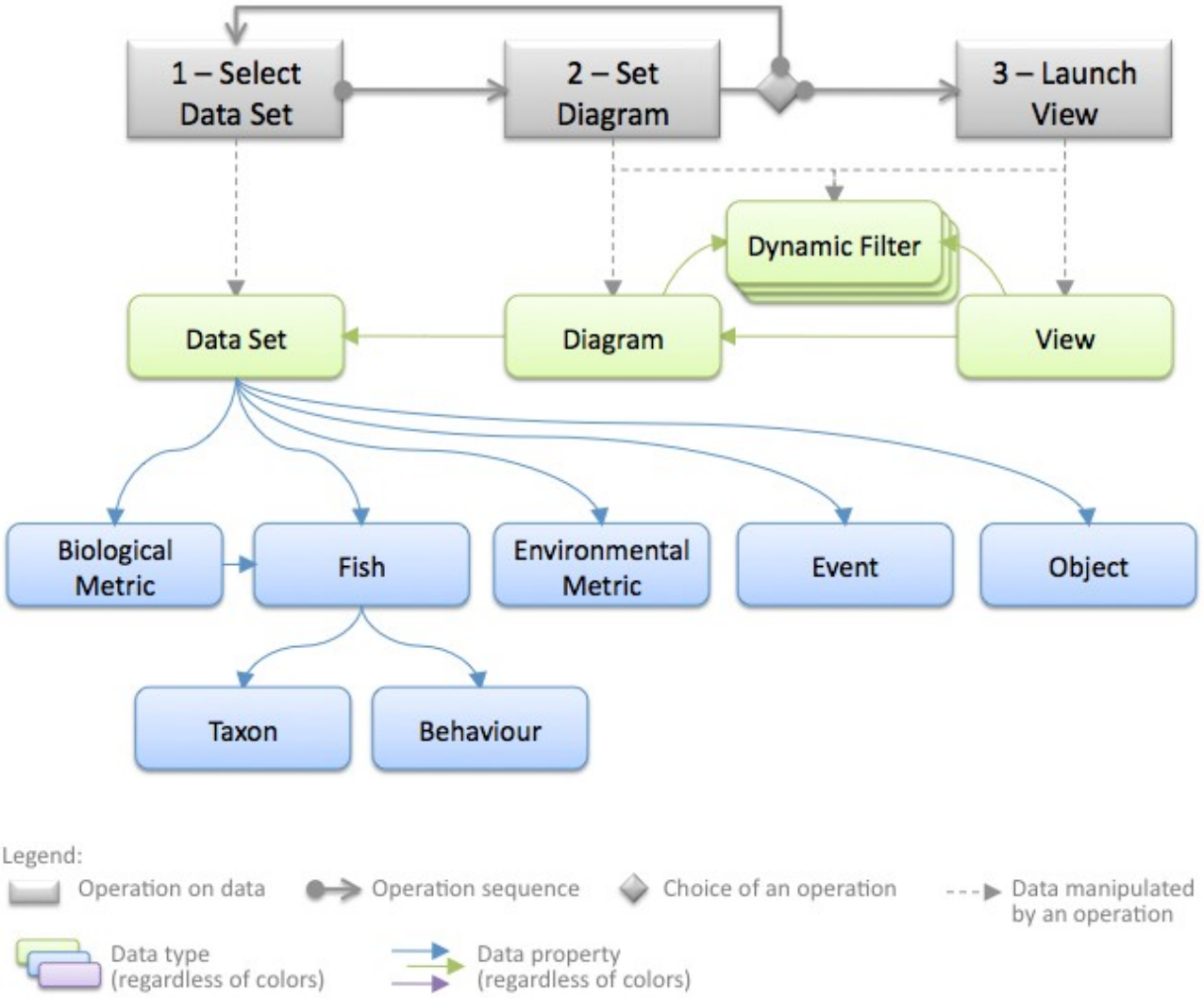

Figure 3-a – Typical data manipulations for creating a personalised data visualisation (Requirements R1, R2)

The Figure 3-b below gives an example of the creation of a custom *Diagram* that illustrates our first user scenario, step 1 (see section 2.1). For this example, the View is a visualisation of the weekly counts of all fish detected in all the monitored locations, for a period of 3 years. First, the user defines the *Data Set* to visualise by selecting the *Biological Metric* of interest, and by indicating the parameters necessary to calculate the values of the metric. In

this example, the metric is the *Overall Abundance* which has only 3 parameters: the timeframe of interest (e.g., from January 2010 to December 2012), the location of interest (e.g., all locations), and the time units in which the metric is calculated (e.g., every week).

On the basis of these parameters, the user has to choose the type of *Diagram* and the *Dynamic Filters*. The user is provided with default choices: a line chart for the *Diagram* and the modification of the time unit and the period of interest for the *Dynamic Filters*.

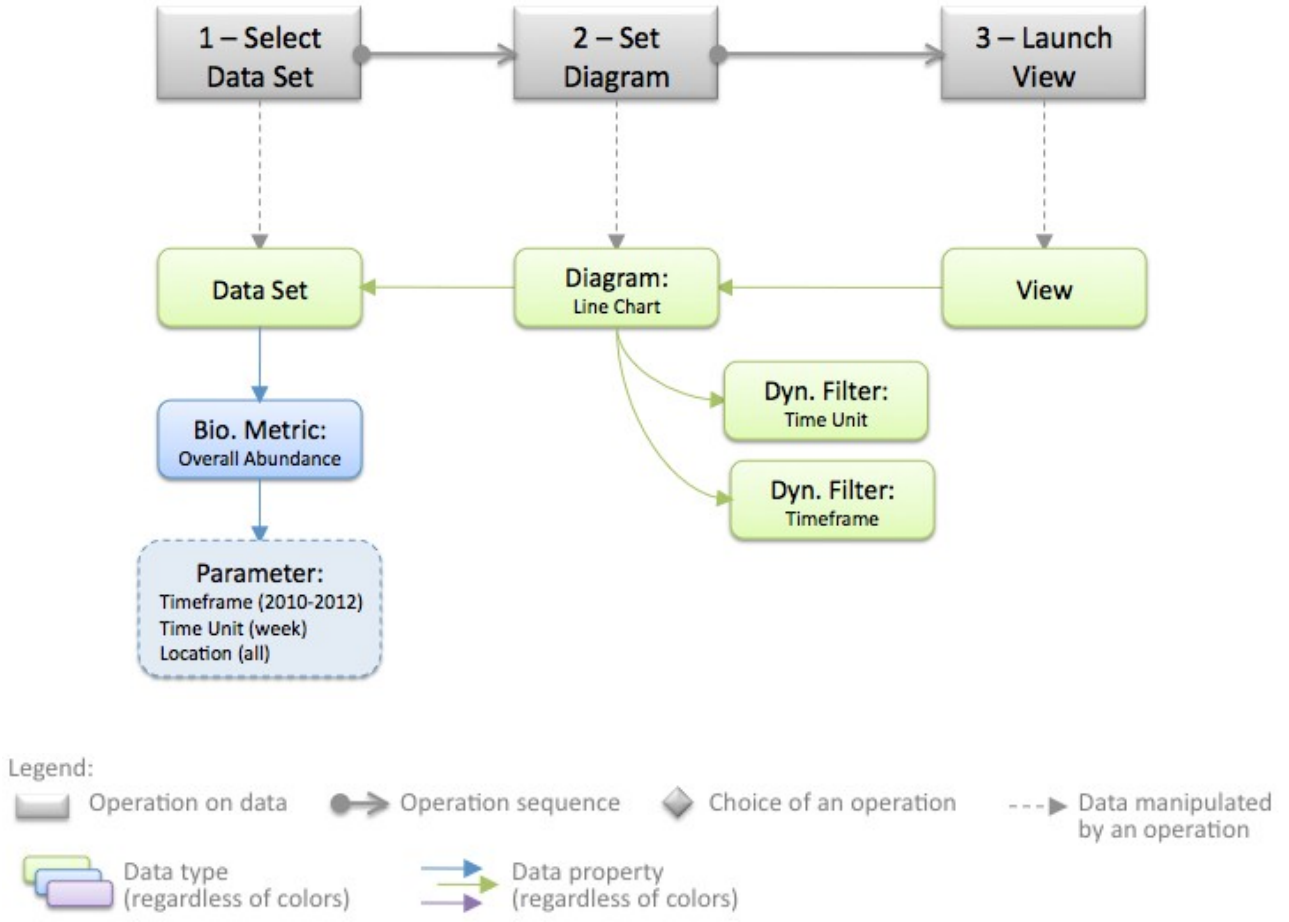

Figure 3-b – Example of data manipulations for creating a personalised data visualisation

e) Yet unprocessed *Data Set*

This section briefly discusses the cases where users request for visualising a *Data Set* that requires data that are not yet available. We consider here 2 cases of data unavailability: the needed video analysis is not done, or the needed computation of *Biological Metrics* is not done. Such additional computations may be necessary for *Data Sets* that contain *Fish* and *Biological Metrics*.

The Figure 4 below illustrates the system responses in such cases, using the examples given above (i.e., in Figures 1 and 2). At step 3, users try to launch a *View* that requires yet unprocessed data. The system warns users that data are yet to be computed, and indicates the estimated response time for the extra computation. Then users can either i) choose to modify the requested *Data Set* (i.e., back to step 1 or 2), or ii) ask to be notified when the extra computations are completed (i.e., step 4).

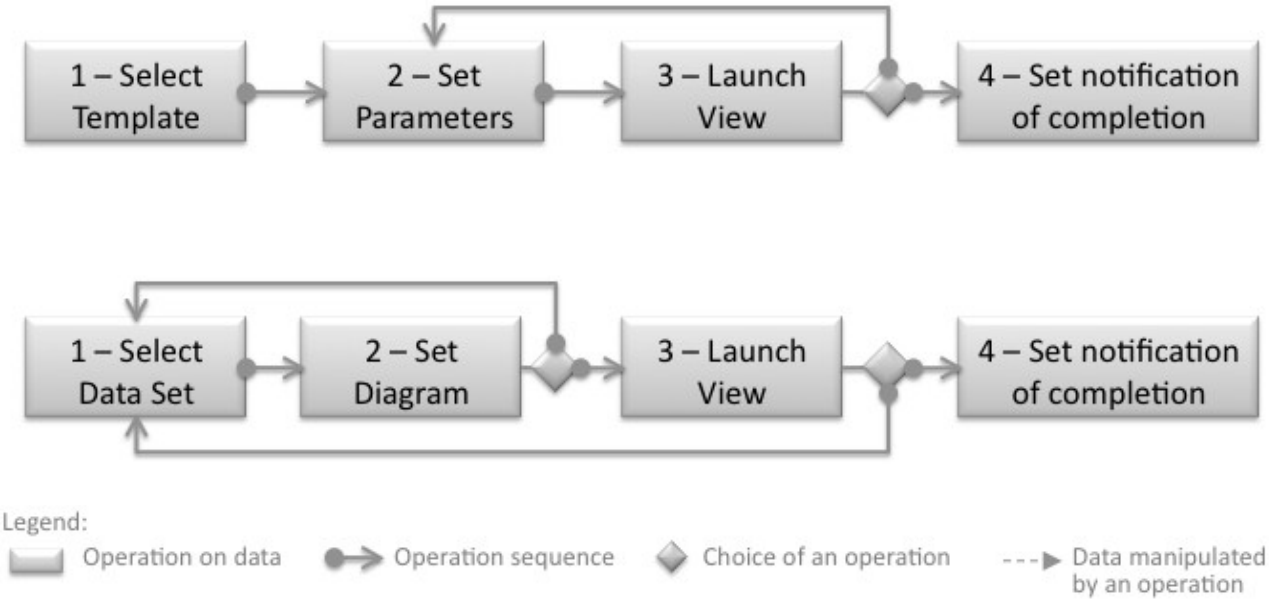

Figure 4 – Typical data manipulations when a requested Data Set needs additional computations (Requirements R1, R2)

These cases may imply interaction with the Workflow Component, particularly regarding its Goal Ontology.

f) Verify analysis

In this section, we focus on users' need to verify the computations done for analysing *Videos*, and for producing *Fish*, *Objects*, and *Biological Metrics*. The Figure 5 below illustrates 4 main operations done for verifying a data analysis. The verifications typically concern a *Data Set* displayed in a *Diagram*, and containing data from automated video analysis. For verifying video data, users are supported with 4 verification views that answer the following questions:

- **Verifying *Biological Metric***: What sets of *Fish* were used to compute the metric? What were the *Fish* selection parameters? What were the other computation parameters? What algorithm or formula is used?
- **Verifying video analysis data**: What *Fish* entities, or other *Objects*, are involved? How certain is their recognition and their description made by computer vision processes?
- **Verifying video analysis processes**: What *Workflow* processed and

produced the *Fish*? What *Programs* were involved? What was the version of their algorithm? What set of *Videos* was analysed?

- **Verifying videos**: What are the images of the recognised *Fish* or *Object*? Can I watch the video clips where they appear? What is the quality of the video involved? What are the missing videos because of unavailability of cameras or other components?

These 4 verification views are available for each *Diagram* of the user interface. For instance, along with each displayed *Diagram*, a menu could provide the list of verification views. In addition, an overview of these technical details is also available for each *Diagram*. In general, these verification views should be provided each time data processed by computer vision *Programs* are displayed (e.g., *Biological Metrics, Fish, Videos)*.

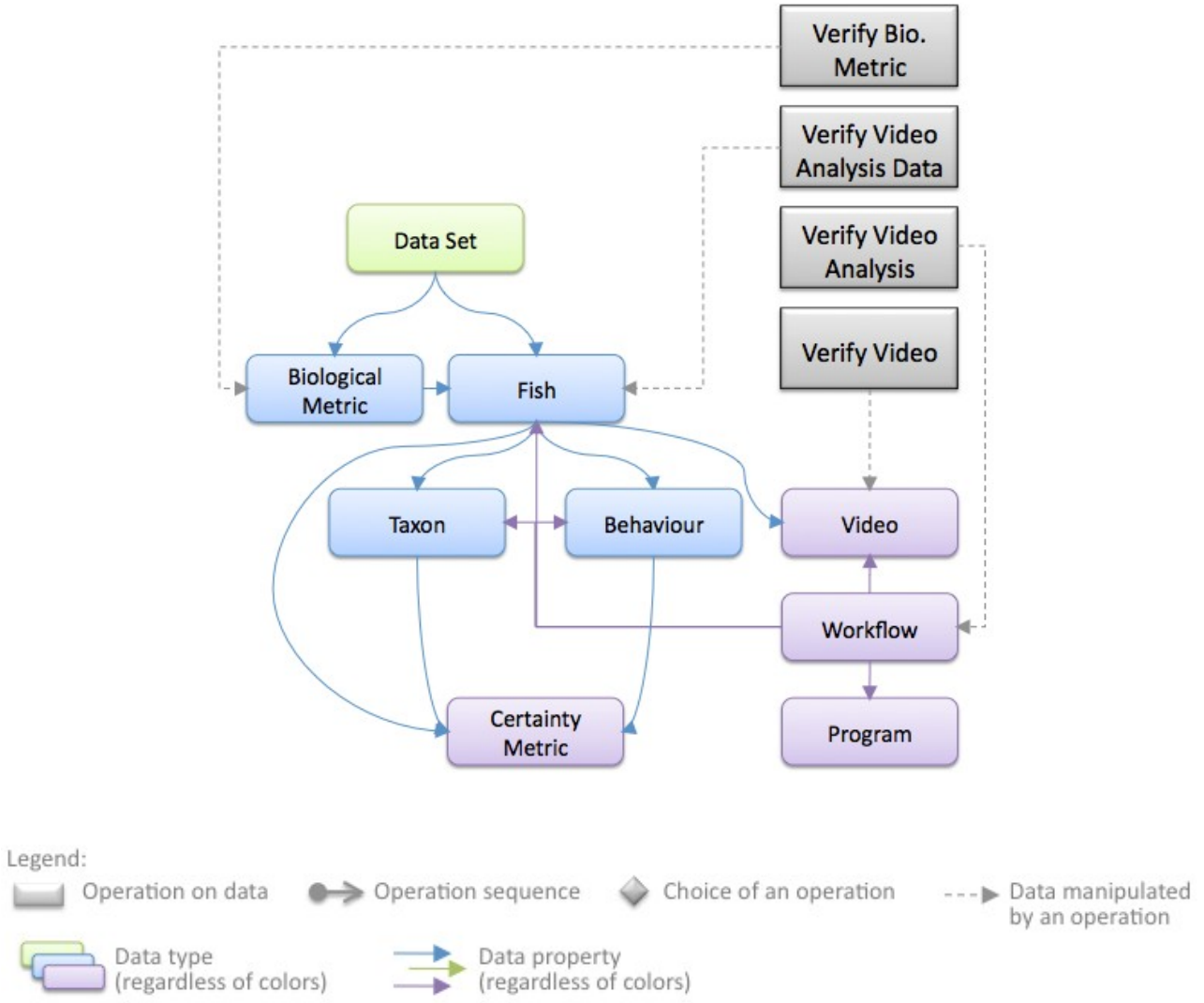

Figure 5 – Typical data manipulations for verifying a Data Set (Requirement R3

The Figures 6 to 9 below highlights the data controlled when verifying each of the 4 technical aspects (*Biological Metric*, video analysis data, video analysis processes, video clips). To simplify the figures, non-fish *Object* are not represented. Non-fish *Objects* would be controlled in the same way as *Fish* objects.

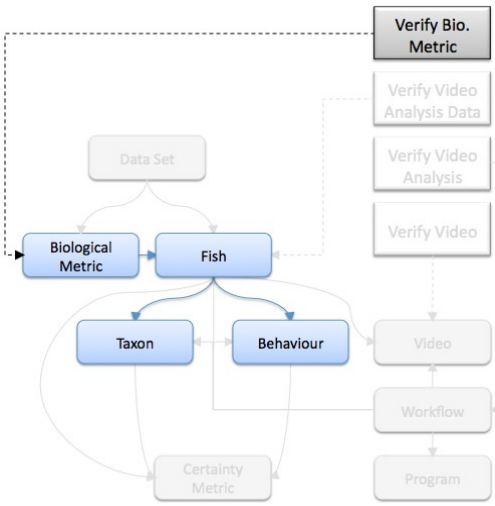

Figure 6 – Data controlled when verifying a Biological Metric

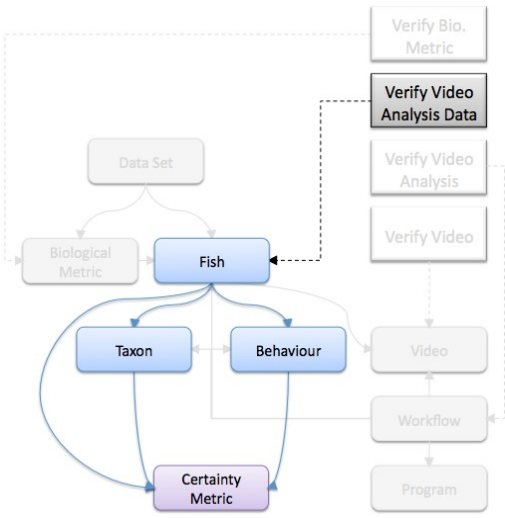

Figure 7 – Data controlled when verifying video analysis data

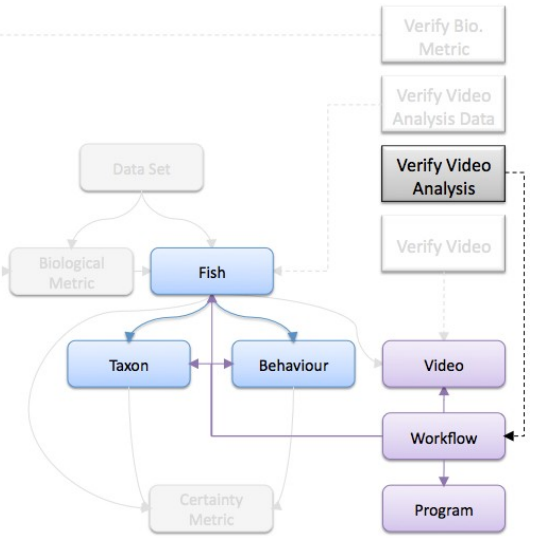

Figure 8 – Data controlled when verifying video analysis processes

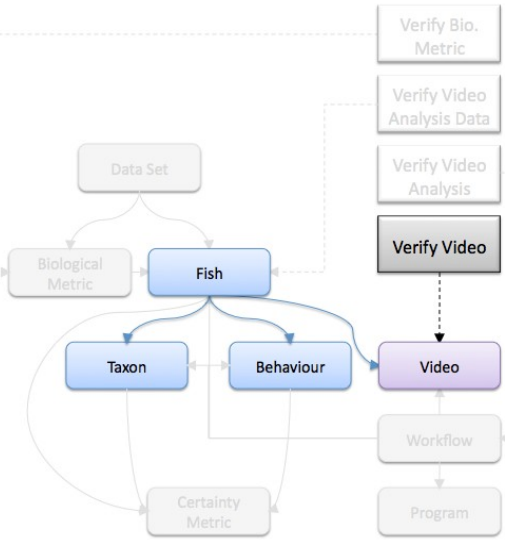

Figure 9 – Data controlled when verifying Videos

The Figure 10-a below illustrates the overall sequence of operations made to verify data analyses. A *verification overview* can be given for a *Diagram*. From that overview, users can access detailed verification views to focus on an aspect in particular. For instance, when overviewing the technical features of a *Diagram*, users are supplied with an overview of all the *Metrics* calculated for the *Diagram*. They can select a metric and verify the video analysis data (i.e.,

the *Fish*) that were used to calculate this metric.

When visualising a detailed verification view (verifying metrics, video analysis data, video analysis processes, and videos), users can explore other technical features and switch to another verification view. Users can also navigate back to the verification overview.

The detailed verification views could also be accessed directly from the *Diagram*, without needing to display an overview first.

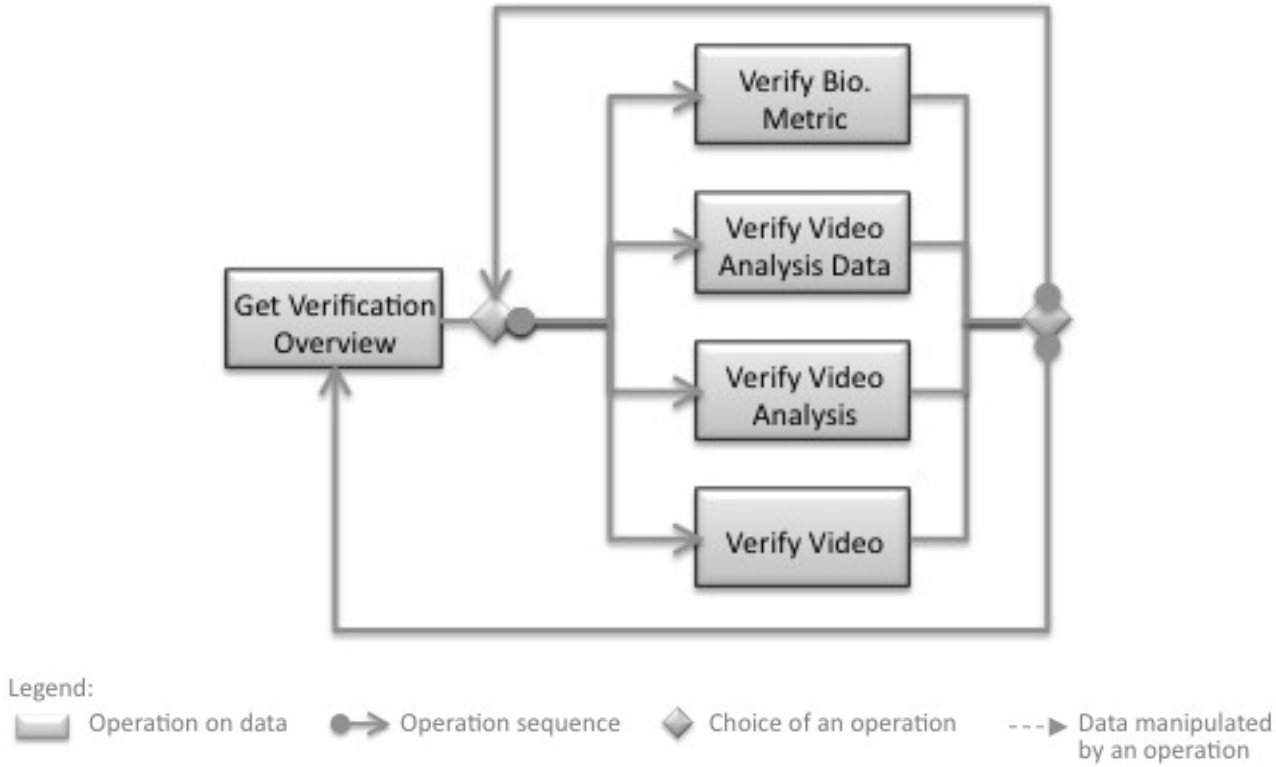

Figure 10-a – Overall verification process

The Figure 10-b gives an example of a verification process that illustrates our first user scenario, steps 3 to 6 (see section 2.1).

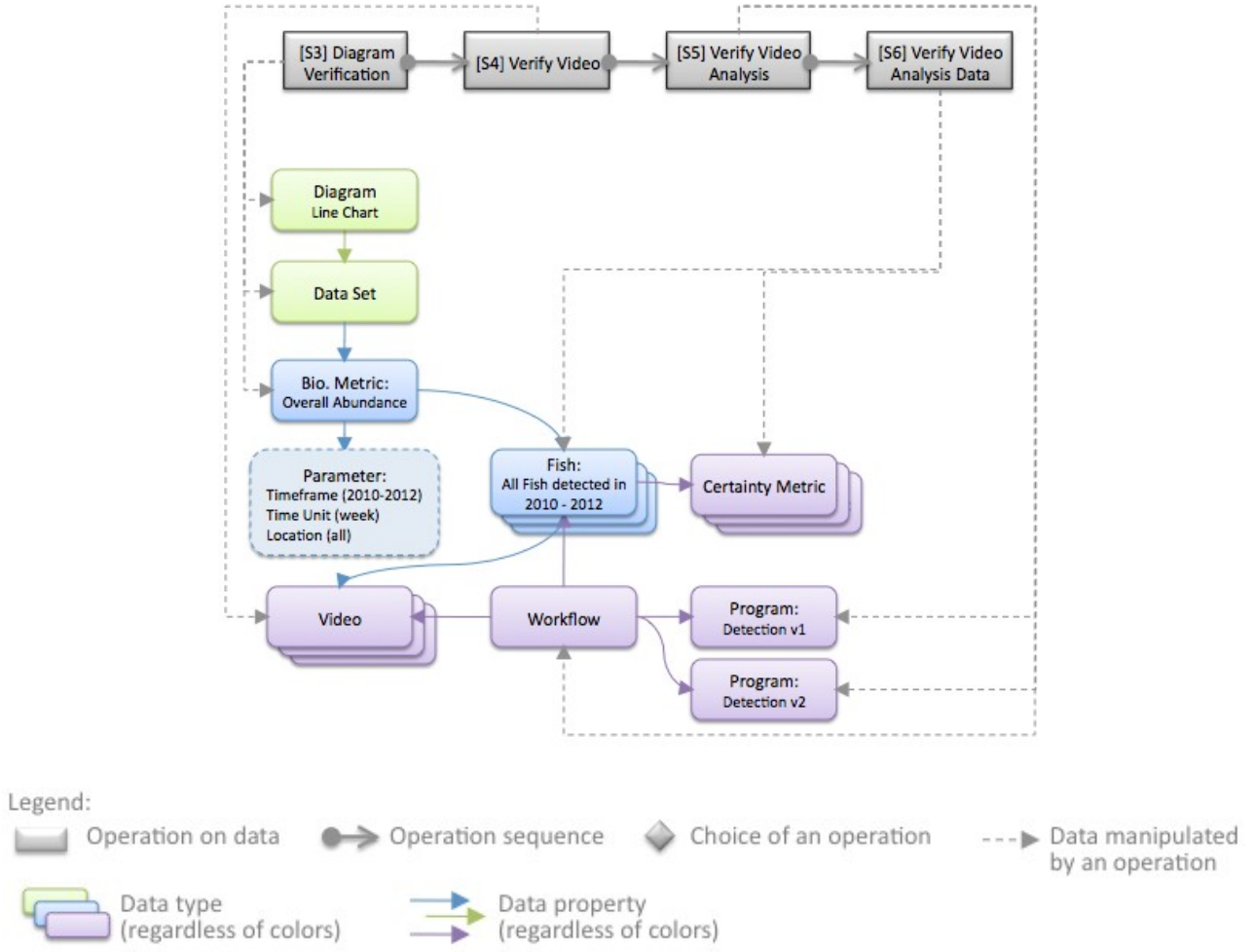

Figure 10-b – Example of a verification process and the controlled data

g) Adapt analysis

Users can modify a large variety of parameters of a data analysis. We organised them into 5 categories:

1. **Representation**: the parameters used to specify the visual display of data (i.e., without requiring any computation of the Data Set to display). For instance, users would control the colours, layout, dimensions, types of diagram (e.g., line chart, bar chart, heat map).
2. **Biological metrics**: the parameters used to specify the calculation of domain-specific measurements (e.g., abundance, growth rate). For instance, users would control the parameters of the measurements (e.g., the timeframe and time unit), or the type of measurement itself (e.g., measure the overall abundance and the species richness using the same data set).
3. **Computer vision data**: the parameters used to select the entities and features that were recognised and described by the computer vision

programs. For instance, users would control the locations and timeframe where entities are detected (e.g., select *Fish* from area A in month M), or the recognition certainty of the selected entities (e.g., select *Fish* with a *Certainty Metric* above a specific threshold).

4. **Computer vision processes**: the parameters used to specify the sequences of programs that recognised entities and features in the videos. For instance, users would control the *Programs* or the *Workflow* that produced the entities (e.g., use *Fish* entities that were produced using a specific set of *Programs*), or the version of computer vision programs (e.g., use *Fish* entities that were produced using a *Program* of *type T* with *version V*).
5. **Video**: the parameters used to select the video clips that were analysed by computer vision processes, and where the recognised entities appear. For instance, users would control the quality of the videos images, or the quality of lightning.

To facilitate the modification of parameters, we will support users with the following mechanisms:

- Users can modify the parameters of the 5 categories described above. Parameters are accessed category per category, and *Diagram* per *Diagram*. The Figure 11 below illustrates the process of accessing parameters of each *Diagram* and of each category. The process of accessing data analysis parameters is purposively similar to the process of verifying data analysis. This similarity is firstly meant for supporting users with a consistent comprehension of the underlying processes that generate the information and the *Views*. It is also meant to let users modify data analysis parameters when needed along the verification task. Conversely, it is also meant to let users verify the data analysis before modifying its parameters.
- If a *Template* was used, users can modify the parameters originally used to instantiate the *Template*. *Views* that are created using a *Template* have a property that links back to the *Template* used and its parameters.
- Users can **save**, **duplicate**, **name** and **comment** a View, a Diagram or a Data Set. These functionalities are meant to keep track of the data analyses done over time.

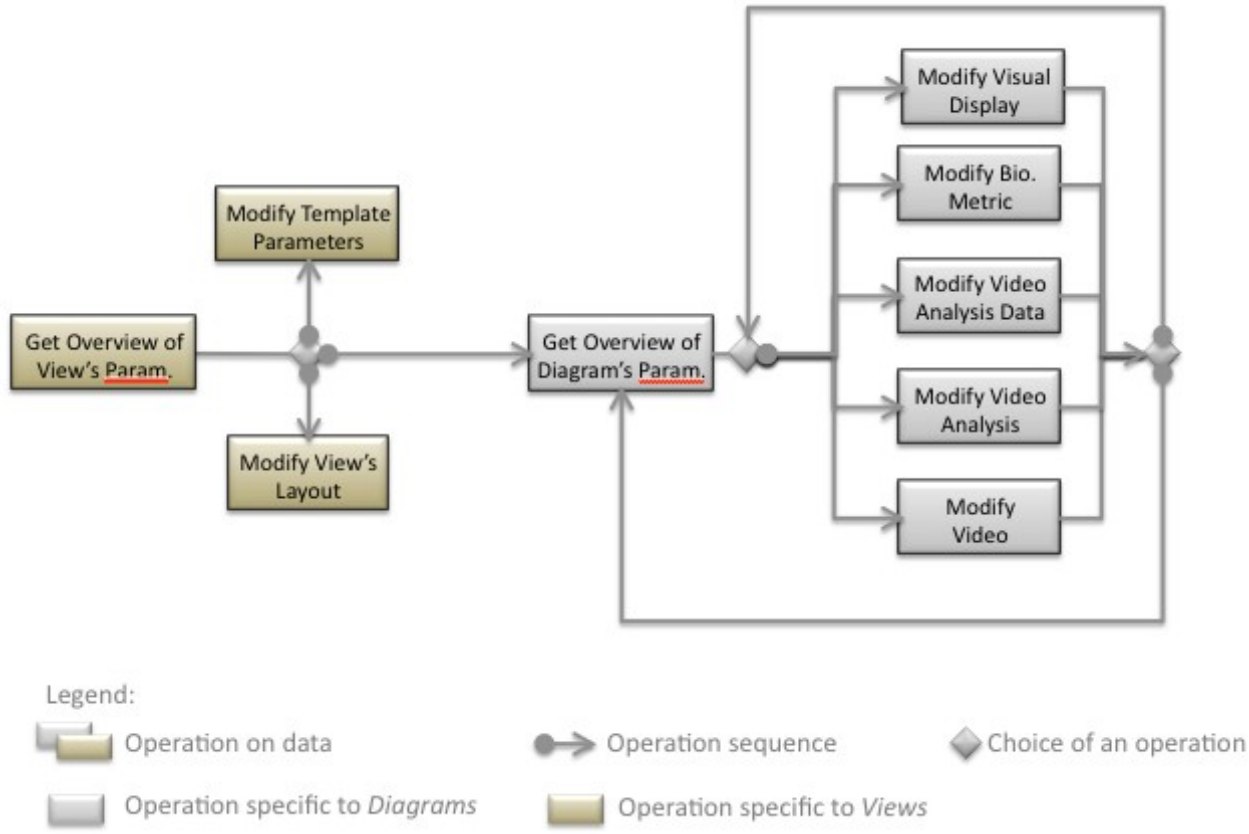

Figure 11 – Overall modification process (Requirement R4)

3.4. Detailed specification of the data types

This section details the domain-oriented data that will be supplied to users, and the functionalities to manipulate these data.We use an object-oriented approach to specify the data types. It consists of defining data as objects with properties and operations. This formulation facilitates the description of i) relationships between data, and ii) basic operations on data. This object-oriented specification serves first for specification purposes. The user interface might not use an object-oriented paradigm, nor a semantic-based or object-oriented implementation. The user interface paradigm and the implementation technique will be decided after further experimentation (described in section 4).

a) View

A *View* is a visualisation of data sets that are displayed using interactive graphical representations (i.e., *Diagrams*).

Properties of *Views* [cardinality is indicated between brackets]:

- Name [1-1]: users can define one descriptive name for the *View*. It briefly expresses the interpretation of the *View*, or how useful is the *View*

for a particular study. It is also a label to designate the *View* in the user interface (e.g., in a list of *Views*). A default name is automatically set when a new *View* is created.

- Comment [0-1]: users can further describe the *View*, its interpretation or any kind of remark, by using one free form text.
- Layout Item [1-*]: a *View* can contains a number of interactive graphical representation of data, called *Diagrams.* A *Layout Item* property specifies a *Diagram* that is contained in the *View*, and the parameters for displaying the *Diagram* in the layout of the *View* (i.e., position, size). Thus the *Layout Item* property is a containers for one *Diagram* ID and a set layout parameters.
- Filter [0-*]: a *View* can have *Dynamic Filters* that allow to modify all the *Diagrams* of the *View*. Users can modify the data sets that are represented in every *Diagram*. For instance, they can modify the timeframe or the location of interest. This would prevent data that do not satisfy the filtering conditions to be displayed. Users can also modify the time units of the *Diagrams*. More details can be found below in the section dedicated to *Filters*.
- Template [0-1]: if a View was created using a Template, the Template used is associated with the View in case users want to modify the Template's parameters.

Operations on *Views*:

- Create, Delete, Save, Duplicate, Name, Comment
- Add a Diagram, Remove a Diagram, Reposition a Diagram, Resize a Diagram
- Add a Dynamic Filter, Remove a Filter, Set a value for a Filter

b) Diagram

A *Diagram* is an interactive graphical representation of a data set.

Properties of *Diagrams* [cardinality is indicated between brackets]:

- Name [1-1]: users can define one descriptive name for the *Diagram*. It briefly expresses the interpretation of data, or how useful is the *Diagram* for a particular study. It is also a label to designate the *Diagram* in the user interface (e.g., in a list of *Diagrams*). A default name is automatically set when a new *Diagram* is created.
- Comment [0-1]: users can further describe the *Diagram*, its interpretation or any kind of remark, by using one free form text.
- Type of Representation [1-1]: A same set of data can be represented in several ways (e.g., line chart, bar chart, map...). Users can modify the

way data are represented by switching the type of graphical representation. Further, every graphical representation need parameters to specify its display (e.g., colours or icons to represent each data set). Thus the *Type of Representation* property is a container for one type of representation and for a set of representation parameters.

- Data Set [1-*]: A *Diagram* can represent one or several sets of data (e.g., a count of fish, a set of events...). More details can be found below in the section dedicated to *Data Sets.*
- Filter [0-*]: a *Diagram* can have *Dynamic Filters* that allow to modify the data sets that are represented. For instance, they can modify the timeframe or the location of interest. This would prevent data that do not satisfy the filtering conditions to be displayed. Users can also modify the time units of the *Diagram*. More details can be found below in the section dedicated to *Filters.*

Operations on *Diagrams*:

- Create, Delete, Save, Duplicate, Name, Comment
- Set the type of Representation, Set Representation Parameter.
- Add a Data Set, Remove a Data Set
- Add a Dynamic Filter, Remove a Filter, Set a value for a Filter
- Verify

c) Data Set

A *Data Set* is a collection of data chosen among the computer vision data (e.g., detected *Objects*, *Fish*, *Species* and *Behaviours*), the *Environmental Metrics* (e.g., water temperature, atmospheric pressure...), the *Biological Metrics* (e.g., counts of fish, counts of species...), or the environmental *Events* (e.g., locations of typhoons...).

Properties of *Data Sets* [cardinality is indicated between brackets]:

- Name [1-1]: users can define one descriptive name for the *Data Set*. It briefly expresses the meaning of the *Data Set*, or how useful are the data for a particular study. It is also a label to designate the *Data Set* in the user interface (e.g., in a list of *Data Sets*). A default name is automatically set when a new *Data Set* is created.
- Comment [0-1]: users can further describe the *Data Set*, its interpretation or any kind of remark, by using one free form text.
- Type [1-1]: a Data Set is a set of either *Biological Metrics*, *Environmental Metrics*, *Events*, *Fish*, or *Objects.* These are the five main types of *Data Set.*

- Selection Criteria [1-*]: the set of data to gather in the *Data Set* is defined by indicating the values of the properties of the desired *Metrics*, *Events*, *Fish* or *Objects*. Properties that can be used to defined the desired *Data Set* are those indicated in the section dedicated to *Metrics*, *Events*, *Fish,* and *Objects*.

Operations on *Data Sets*:

- Create, Delete, Save, Duplicate, Name, Comment
- Set Data Type, Add a Selection Criteria, Remove a Selection Criteria, Set a Selection Criteria.

d) Dynamic Filter

A *Dynamic Filter* is a specification of the data displayed in *Diagrams*. A *Dynamic Filter* allows users to refine the data displayed in one *Diagram* (i.e., a *Diagram's Filter*) or in every *Diagram* of a *View* (i.e., a *View's Filter*). For instance, while inspecting a *View*, users can interactively redefine the time unit or the timeframe of interest. The usage of *Dynamic Filter* follows the concept of dynamic queries introduced by by B. Shneiderman [1].

Properties of *Dynamic Filters* [cardinality is indicated between brackets]:

- Property to Filter [1-1]: this defines the property for which users want to apply a selection criteria, by defining the desired values (i.e., se *Condition* below). The *Property to Filter* can be any property of *Data Sets*: timeframe, location, time unit, *Taxon*, *Behaviour*, detection certainty, type of *Metric*, abundance threshold...
- Condition [1-*]: A *Condition* of a *Filter* defines a desired value of the *Property to Filter*. Desired value are defined by i) the operator that verifies the condition (i.e., equals, not equals, superior, inferior), and by ii) the reference value. Thus the *Condition* property is a containers for one operator and one reference value.

Operations on *Dynamic Filters*:

- Create, Delete
- Set the Property to Filter, Add a Condition, Remove a Condition, Modify a Condition

e) Template

A *Template* is a model of a *View* that allow to quickly create a standard, frequently used *View* by indicating only a few parameters. For specifying a View, a *Template* comprises predefined parameters, and user-defined

[1] Schneiderman B., Dynamic queries for visual information seeking. In IEEE Software, pages 70-77, 1994.

parameters. *Templates* allow to create a standard, frequently used *View* by indicating only a few user-defined parameters. For instance, *Templates* allow to quickly create elaborated *Views* displaying several *Data Sets* and *Biological Metrics* sharing the same parameters.

Properties of *Templates* [cardinality is indicated between brackets]:

- Name [1-1]: *Templates*' names are predefined.
- Parameter [0-*]: To create a new standard *View*, users launch a *Template* and define the needed parameters. Some Template might not need any parameter (e.g., the default overview for fish detected today, or this month). Most of the *Templates* need custom parameters, among: timeframe, location, time unit, *Taxon*, *Behaviour*, detection certainty, type of *Metric*...

Operations on *Templates*:

- Launch Template, Set a Parameter

f) Biological Metric

A *Biological Metric* is a measurement used by biologists to study fish populations (e.g., abundance, growth rate). The Biological Metrics are those defined in the Deliverable 2.1, particularly in its Appendix VI.

Properties of *Biological Metrics* [cardinality is indicated between brackets]:

- Type [1-1]: the type of metric is either: Overall Abundance, Overall Growth Rate, Abundance, Growth rate, Relative Abundance, Abundance Level, Composition of Abundance Level, Species Richness, Species Richness Growth, Species composition, Species Composition Change, Behaviour occurrence, Behaviour Occurrence Growth.
- Name [1-1]: *Biological Metrics*' names are predefined.
- Description [1-1]: *Biological Metrics*' descriptions are predefined. For instance, they briefly explain the way metrics are calculated, how to choose the parameters and how to interpret the results.
- Parameter [2-*]: Parameters must be defined to select the data to use for the calculation of the metric (e.g., a species must be defined to calculate its Abundance). Parameters of *Biological Metrics* are defined in the Deliverable 2.1, Appendix VI, section 3. All metrics have two mandatory parameters: the Timeframe and the Location of interest.

Operations on *Biological Metrics*:

- Launch Metric, Set Type, Set Parameter
- Verify, Use for an Analysis

g) Environmental Metric

An *Environmental Metric* is a measurement used to monitor meteorological conditions (e.g., temperature, pressure), and other environmental conditions (e.g., pollution). Environmental sensors or meteorological agencies can supply regular measurement of environmental parameters. Our partners, and potential users, from Academia Sinica are already monitoring environmental conditions with measurements such as water temperature or atmospheric pressure. Users will visualise these data, and summarise them using larger time units (e.g., a daily average of the temperature measured every 30 minutes).

Properties of *Environmental Metrics* [cardinality is indicated between brackets]:

- Type [1-1]: the type of metric is for instance Water Temperature, air Temperature, Pressure, or Current Velocity.
- Name [1-1]: *Environmental Metrics*' names are predefined.
- Description [1-1]: *Environmental Metrics*' descriptions are predefined. For instance, they briefly explain the way data are collected, and how to use and interpret the metrics.
- Parameter [2-3]: Two parameters are mandatory to produce a *Environmental Metric*: the Timeframe and the Location of interest. In addition, users can also define a Time Unit that will be used to calculate *Environmental Metrics* for each Time Unit within the Timeframe of interest. Thus the *Parameter* property is a container for one Timeframe, one Location, and zero to one Time Unit.

Operations on *Environmental Metrics*:

- Launch Metric, Set Type, Set Parameter
- Use for an Analysis

h) Event

An Event represent a thing that happened and has an environmental importance. An Event is meant to define a period and location of interest to be considered for video data analysis. It also describes the type of thing that happened.

Properties of *Events* [cardinality is indicated between brackets]:

- Type [1-1]: the type of *Event* is either: Habitat Modification, Pollution, Natural Event.
- Name [1-1]: users can define one descriptive name for the *Event*. A default name is automatically set when a new *Event* is created.

- Comment [0-1]: users can further describe the *Event*, the expected impacts or any kind of remark, by using one free form text.
- Location [1-1]: a container for the coordinates defining the area of occurrence of the *Event.*
- Timeframe [1-1]: a container for the beginning and ending timestamps in which the *Event* happened.

Operations on *Events*:

- Create, Delete, Save, Duplicate, Name, Comment
- Set Type, Use as Metric Parameter, Use for an Analysis

i) Fish

A *Fish* represents an entity that was filmed, and that was recognised as a fish by automated video analysis. A Fish is described by three main computer vision data: the taxonomic classification (i.e., species, genus, family), the observed behaviour and the certainty of the above recognitions.

Properties of *Fish* [cardinality is indicated between brackets]:

- Taxon [0-1]: If recognised, the taxonomic classification of the Fish is represented by the fish *Species*, *Genus* and *Family*. The certainty of the classification is also indicated. Thus the *Taxon* property is a container for one *Species ID*, one *Genus ID*, one *Family ID* and one certainty score.
- Behaviour [0-*]: If recognised, the activities of the fish are indicated. The certainty of the behaviour recognition is also indicated. Thus the *Behaviour* property is a container for one *Behaviour ID* and one certainty score.
- Certainty [1-1]: A score combining the certainty of fish detection in each images, and fish tracking over several images.
- Location [1-1]: a container for the coordinates defining the location where the *Fish* was filmed (i.e., the location of the *Camera*).
- Timeframe [1-1]: a container for the beginning and ending timestamps in which the *Fish* was filmed (i.e., timestamp of he first and last images).
- Video [1-1]: the ID of the *Video* in which the *Fish* appeared.

Operations on *Fish*:

- See Video, Use for an Analysis, Verify

j) Object

An *Object* represents an entities that was filmed, and that was recognised as a

non-fish object by automated video analysis.

Properties of *Objects* [cardinality is indicated between brackets]:

- Type [0-1]: The type of an *Object* indicates what kind of entity was recognised (e.g., octopus, garbage..).
- Certainty [1-1]: A score combining the certainty of object detection in each images, object tracking over several images, and the recognition of the *Type* of *Object*.
- Location [1-1]: a container for the coordinates defining the location where the *Object* was filmed (i.e., le location of the *Camera*).
- Timeframe [1-1]: a container for the beginning and ending timestamps in which the *Object* was filmed (i.e., timestamp of he first and last images).
- Video [1-1]: the ID of the *Video* in which the *Object* appeared.

Operations on *Objects*:

- See Video, Use for an Analysis, Verify

k) Certainty Metric

A *Certainty Metric* is a score representing the accuracy of a data produced by computer vision *Program.* It is evaluated through video analysis *Workflows* that recognised a *Fish* entity, an *Object*, a *Behaviour,* or a *Taxon* of a *Fish*. *Certainty Metrics* are elaborated on the basis of detailed technical evaluations of i) prediction of the chance of errors (e.g., false negative, false positive...), and ii) the quality of various features of video materials. The Certainty Metrics given here combine and synthesise these error predictions and quality scores into one value. These detailed error predictions and quality scores could eventually be accessed for specific usages. This is discussed in section 4.3.

Properties of *Certainty Metrics* [cardinality is indicated between brackets]:

- Type [1-1]: The type of *Certainty Metrics* indicates what kind of entity it describes (i.e., either *Fish*, *Object*, *Taxon* or *Behaviour*).
- Value [1-1]: A score summarizing the detailed evaluations of the chances of error and of the quality of features of video materials.

Operations on *Certainty Metric*:

- See Workflow

l) Program

A *Program* is a software component that was used to automatically analyse the videos.

Properties of *Programs* [cardinality is indicated between brackets]:

- Type [1-1]: the type of a *Program* is either: Detection, Tracking, Description, Recognition, Clustering.
- Version [1-1]: *Programs* can be modified and updated. This property indicates each version of *Programs.*
- Comment [0-1]: eventually, users can indicate any remark concerning, for instance the modifications made in an update.

Operations on *Objects*:

- See analysed Video, See all Versions, Use for an Analysis

m) Workflow Task

A *Workflow* represents a sequential usage of Programs that has produced a set of data (i.e., a set of *Fish* or *Objects*). For any set of Fish or Objects, users can visualise which Programs are involved in producing these data. Thus a *Workflow* contains references for all *Programs* used. Each set of *Fish* or *Object* have its own set of Programs involved. Thus *Workflow* entities are defined on-demand for each set of *Fish* or *Objects*. The sequence of Program usage is assumed to be fixed (e.g., Detection then Tracking, then Description, then Recognition).

Properties of *Workflows* [cardinality is indicated between brackets]:

- Detection [1-*]: each *Program* of type "Detection" involved (i.e., each version of Fish Detection component).
- Tracking [0-*]: each *Program* of type "Tracking" involved (i.e., each version of Fish Tracking component).
- Description [0-*]: each *Program* of type "Description" involved (i.e., each version of Fish Description component).
- Recognition [0-*]: each *Program* of type "Recognition" involved (i.e., each version of Fish Recognition component).
- Clustering [0-*]: each *Program* of type "Clustering" involved (i.e., each version of Fish Clustering component).
- Video [1-*]: *Videos* that were processed by the *Programs* involved.
- Fish [0-*]: *Fish* entities, including their *certainty* property, that were produced by the Detection *Program.*
- Behaviour [0-*]: *Behaviours*, including their *certainty* property, that were recognised by the *Workflow.*
- Taxon [0-*]: the taxonomic classifications, i.e. the sets of Species, Genus and Family IDs, including their *certainty* property, that were recognised by the *Workflow.*

- Object [0-*]: non-fish entities, including their *certainty* property, that were produced by the Detection *Program.*

Operations on *Workflows*:

- See analysed Video, Modify Programs, Modify Videos

n) Video

A *Video* is a 10 minute video clip filmed by underwater cameras, and cut into a 10 minute excerpt for storage and video analysis matters.

Properties of *Videos* [cardinality is indicated between brackets]:

- Camera [1-1]: the device that produced the video.
- Location [1-1]: a container for the coordinates defining the location where the *Video* was filmed (i.e., the location of the *Camera*).
- Timeframe [1-1]: a container for the beginning and ending timestamps in which the *Video* was filmed.
- Light Condition [0-1]: an indication of the quality of light exposure that can impact the quality of automated video analysis.
- Water Clarity [0-1]: an indication of the quality of water transparency that can impact the quality of automated video analysis.
- Lens Cleanliness [0-1]: an indication of the presence of algae or dirt on camera lens, which can impact the quality of automated video analysis.

Operations on *Workflows*:

- See Video, See identified Entities, Use for an Analysis

4. Implementation objectives

This section discusses the roadmap and motivations for researching and implementing the user interactions for the Fish4Knowledge system. We identified two research directions that carry the primary Human-System interaction issues and research opportunities we will address. These directions are discussed in sections 4.1 and 4.2. We then discuss other issues of interest in section 4.3.

4.1. Multi-layered analysis

We consider that layers of data consist of the successive, i.e. layered, computational processes that produced the information in use. By *multi-layered analysis*, we mean data analysis that integrates the verification of the underlying computational processes. Users require the control of data computations, and their uncertainties. But users are not computer vision experts. This is the motivation for researching accountable and comprehensive

data analysis that reduces knowledge gaps and information loads.

Our first research direction is to investigate a layered access to the successive processes involved. As described in section 3.3.f, specifically Figures 5 to 9, four layers of control concern the underlying computational processes. In addition, users can also control the visual representation of data. Thus we plan to experiment with five layers of control:

1. **Representation**: the visual encoding techniques for displaying data. For instance, users would control the colours, layout, dimensions, types of diagram (e.g., line chart, bar chart, heat map).
2. **Biological metrics**: the calculation of domain-specific measurements (e.g., abundance, growth rate). For instance, users would control the data used for the measurements (e.g., the fish entities), or verify the formula of the metrics.
3. **Computer vision data**: the entities and features that were recognised and described by the computer vision programs. For instance, users would control the certainty of the recognition and description of the entities (e.g., certainty species recognition, certainty of behaviour recognition).
4. **Computer vision processes**: the sequences of programs, i.e., the workflow tasks, that recognised entities and features in the videos. For instance, users would control the version of computer vision programs.
5. **Video**: the video clips that were analysed by computer vision processes, and where the recognised entities appear. For instance, users would control the quality of the videos, and watch images or video excerpts of fish entities.

As described in section 3.3, when users verify data analyses, they might also need to modify the technical parameters. Thus the five layers listed above should also supply modification functionalities. Another Human-Computer Interaction challenge is preserving the understandability and the usability of the tool. Moreover, our research challenge is even to improve understandability and usability. Thus we will research means to support users with i) a precise understanding of the underlying computational processes, and ii) integrated expert user interactions for controlling the detailed computational parameters.

We plan to investigate, and eventually enhance, several interaction techniques and user interface paradigms such as zoomable interfaces, object-oriented interfaces, 3D-4D interfaces (this technique will also be investigated for data visualisation within *Diagrams*).

4.2. Multi-faceted analysis

We consider that a facet of data is a specific interpretation that is elaborated by gathering purposive information. By *multi-faceted analysis*, we mean data

analysis that allow users to gather their information of interest and to interpret them in their own way. Interpretation of data depend on specific user study, and the very same data set can carry different meanings depending on user studies and interests. Further, data interpretation and user interests evolve through time. There is the motivation for researching evolutive and flexible data analysis that allows to express both domain-specific and user specific interpretations.

We particularly consider 3 levels of interpretation:

- **Single piece of data** (e.g., one query result): for instance, a single count of fish can denote either a depleted species, an improbable value, or a reference threshold.
- **Set of data** (e.g., correlative or comparative set of query results): for instance, users can calculate the same metric over time or over areas, or comparing different metrics. This allows various observations and interpretations of trends.
- **Overall study** (e.g., set of query results plus prior knowledge and external information): for instance, users working environment comprises other source of information, and specific working processes. In this environment, the Fish4Knowledge tool plays a specific role and the data interpretation is elaborated along with other sources of information and follows the overall focus of specific studies.

Regarding the levels and the variety of interpretations, we will research the ways to gather data and elaborate purposive units of information. In that direction, we will particularly focus on four information artefacts specified above: *Data Sets, Diagrams, Views, Templates*. To manipulate these information artefacts, we plan to experiment with the integration of basic low-level functionalities such as: Gather, Name, Comment, Reuse, Duplicate, Modify.

Following Ben Schneiderman's work on basic data types and functionalities for information visualisation[1], we will also investigate basic data types and functionalities for evolutive information interpretation.

4.3. Future refinements

The two directions for experimentation discussed above will allow the implementation of a first prototype of the user interface that integrates all the components and functionalities of the Fish4Knowledge system. We identified several directions for refining this first prototype:

- **Automated detection of outlying situations**: it is possible to detect abnormal or exceptional values of biological metrics. For instance, this

[1]Schneiderman B., The eyes have it: a task by data type taxonomy for information visualizations. In IEEE Symposium on Visual Languages, pages 336-343, 1996.

would allow queries such as "*When and where is there a large change in populations of species X?*". This would require further user studies to define how to target the interesting sets metrics to automatically monitor, how to define the threshold for metric values that denote outlying situations, and how to define the user interface functionalities and their parameters.

- **Expert usage of fish clustering**: The Clustering component allows to elaborate clusters of similar fish, on the basis of any detectable features. Users could benefit for expert usage of the Clustering component. For instance they could query for clusters of fish that behave similarly in the same periods or locations, or that have similar growth rate patterns. This would require further user studies to target the interesting features or clusters, and to define the related user interface functionalities and their parameters.
- **Creation of custom Template**: user-defined templates would be useful when frequent data analysis are not supported by existing *Templates*.
- **Named entities for defining areas and habitats** of interest: defining locations of interest is a frequent data manipulation for many user interactions and data analyses. Allowing user-defined entities representing areas (i.e., sets of camera's locations) would facilitates this parameter definition, and make it more meaningful. For instance, an area entity could be used as a parameter for calculating biological metrics (e.g., similarly Events can be used as a parameter for defining both the timeframe and the location parameters). Further, user-defined entities could allow to describe the types of habitats of the monitored areas.
- **Creation of custom metrics**: all metrics of interest are not supported by default, but it could be possible to allow the creation of custom metrics, i.e. additional computations that use the data our tool can supply. We could study means for users to combine existing metrics, or use custom program or libraries to calculate their metrics of interest.
- **Expert usage of certainty metrics**: the single certainty metric associated to *Fish*, *Objects*, *Taxon* and *Behaviours* is calculated by combining several metrics that evaluate a specific quality of feature detection, or a specific error probability. For instance, evaluations of the quality of fish contour and of the quantity of algae are both used to calculate the certainty of species detections. The detail of certainty metrics might be checked by expert users, and eventually used as parameters for filtering and analysing data.

