RESPONSEABLE

# The ResponSEAble Knowledge Base

*User and Technical Manual*

*Work Package 1*

*Deliverable 1.2*

*March 2019*

Project coordinator:

Project beneficiaries:

This project has received funding from the European Union's Horizon 2020 Framework Programme for Research and Innovation (H2020-BG-2014-1) under grant agreement No. 652643.
This publication/multimedia product/presentation reflects the views of the author, and the European Union cannot be held responsible for any use which might be made of the information contained therein.

# The ResponSEAble Knowledge Base

***Authors***

Caroline Brennan, NUIG

Owen Molloy, NUIG

# Table of Contents

# 1.Introduction

One of the challenges in the ResponSEAble project was how to organise our knowledge about specific topics, or Key Stories, such as Sustainable Fisheries, Ocean Plastics, Invasive Species, etc.. Once we had decided to adopt the DAPSIWR framework to model human-ocean systems, we knew we needed a way to store information and knowledge about these systems. For example, the causal links that exist between Drivers and Activities, or the Actors which are involved in specific value chain activities. Knowledge might take a wide range of forms. For example, scientific journal articles about the causal links between specific human activities, ecosystems pressures and impacts, magazine articles on the impact of plastic microbeads, or videos of sustainable fishing practices. So, we needed a means of mapping both the causal links that exist in a Key Story, using the DAPSIWR framework, as well as the knowledge relating to the nodes and links in that map. Having captured that information, we would then be able to use it in developing ocean literacy tools, as well as analysing it for gaps in knowledge on specific topics or causal links. So, the idea of the Knowledge Base (KB), as a repository of Key Story maps and knowledge was conceived.

The Knowledge Base is an online tool which allows users to create DAPSIWR maps of any ocean or environmental topics, and to link knowledge items parts of the story map or the story as a whole. Tools have been created to facilitate the import of knowledge into the Knowledge Base, and to allow easy querying and reporting of the knowledge it contains. There is also an online KB Viewer to allow easy, interactive viewing of Key Stories and their knowledge.

The Knowledge Base is aimed at a wide range of users. It could be used by, for example, students doing ocean / environmental projects using the DAPSIWR framework to map, understand and publish their system understanding. It could be used by ocean literacy / domain experts to capture comprehensive knowledge on a topic, and use the querying / reporting functionality to quiz the data. It could be used by policy makers as a knowledge resource to deepen their understanding of the causal complexity of Key Stories.

The level of information technology literacy needed to use the systems if low – anyone who can use a browser and drag-and-drop functionality can use the system. The complexity of the models / maps produced would depend on the level of knowledge and expertise of the users.

The system could be used, in its current state, to create DAPSIWR maps of any environmental topic. Other frameworks could also in theory be used with some modification of the software.

If you are interested in using the Knowledge Base or have any questions about it, please contact:

Dr. Owen Molloy, National University of Ireland, Galway.

Tel: 00 353 91 493330

Email: owen.molloy@nuigalway.ie

# 2. Key Stories in ResponSEAble

Existing knowledge on the marine environment (data and information) is very much focused on marine science and education. Clear definition of the roles and dependencies of the various stakeholders in marine systems (e.g. domain experts, policy makers, producers, consumers) is often missing or poorly understood. Stakeholders roles in effecting change is rarely understood in terms of the overall SYSTEM, often limited to best practices for specific sectors. Overall systems thinking and knowledge on how to "do better" within the context of a complex system of causal processes is difficult to achieve and very difficult to communicate and pass on to others. To support changes in Ocean Literacy and hence, hopefully, behaviour, it is not enough to just have the knowledge about the state of environment. We also need knowledge about the complexity of connections and linkages between different drivers and pressures that cause degradation in the state of ecosystem goods and services, and most importantly, connection with impact and response.

In order to overcome these issues and to assist in the structuring of Ocean Literacy knowledge and interventions, we adopted the term 'Key Story' to represent specific topics on the human-ocean relationship, such as Sustainable Fisheries or Coastal Mass Tourism. The modelling framework we chose to map and convey these Key Stories is the DAPSIWR framework (see Figure 2-1).

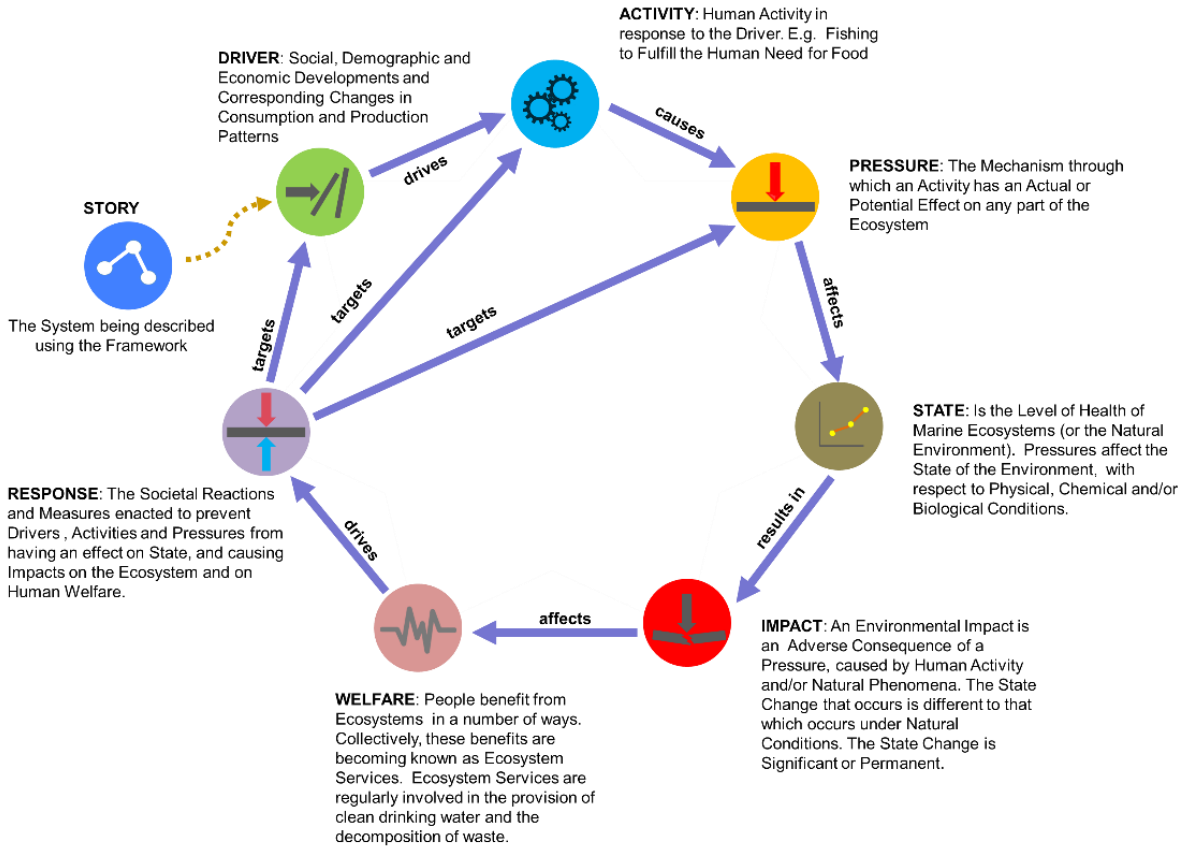



*Figure 2-1: DAPSIWR Framework*

DPSIR is a causal framework for describing the interactions between society and the environment. We have adapted and modified this framework to support modelling of the human-ocean relationship in a more complete fashion, including (Value Chain) Activities and Ecosystem Services.

We also recognised the need to augment this framework with the notion of Actors – those roles which humans play, primarily with respect to the Activities in the causal map. In order to accurately understand and model the system under examination, it is necessary to

gather knowledge and information on individual Drivers, Pressures etc., but also on the Causal links between the components of the system. The process involved in gathering and representing this story 'knowledge' is shown in Figure 2-2.

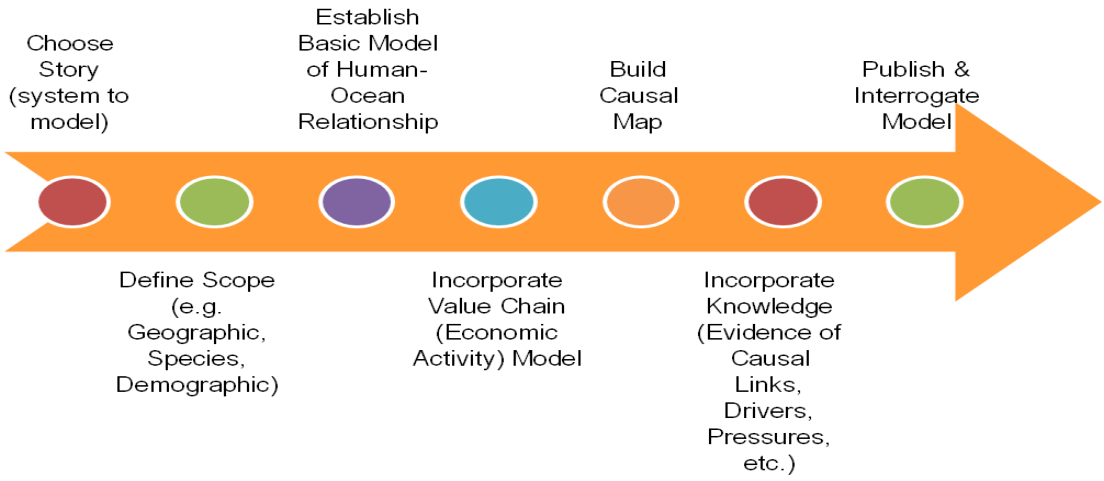



*Figure 2-2: Story Development Process*

# 3. About the ResponSEAble Knowledge Base

The browser-based Knowledge Base Editor facilitates the creation of causal maps modelled using the DAPSIWR[1] framework. For the ResponSEAble project, Key Stories, such as sustainable fisheries, microplastics and coastal tourism, were created using the Knowledge Base.

Each DAPSIWR Story tells a story of chains of causes and effects. Each node (circle) is a type of Driver, Activity, Pressure, State, Impact, Welfare or Response. The connecting arrows show the effects of one node on another. Actors, such as Fishermen or the General Public, can be added to appropriate nodes and relationships in the Story (e.g. see Figure 3-1).

[1] Patrício, J., Elliott, M., Mazik, K., Papadopoulou, K.-N. & Smith, C. J. 2016. DPSIR—Two Decades of Trying to Develop a Unifying Framework for Marine Environmental Management? Frontiers in Marine Science, 3.

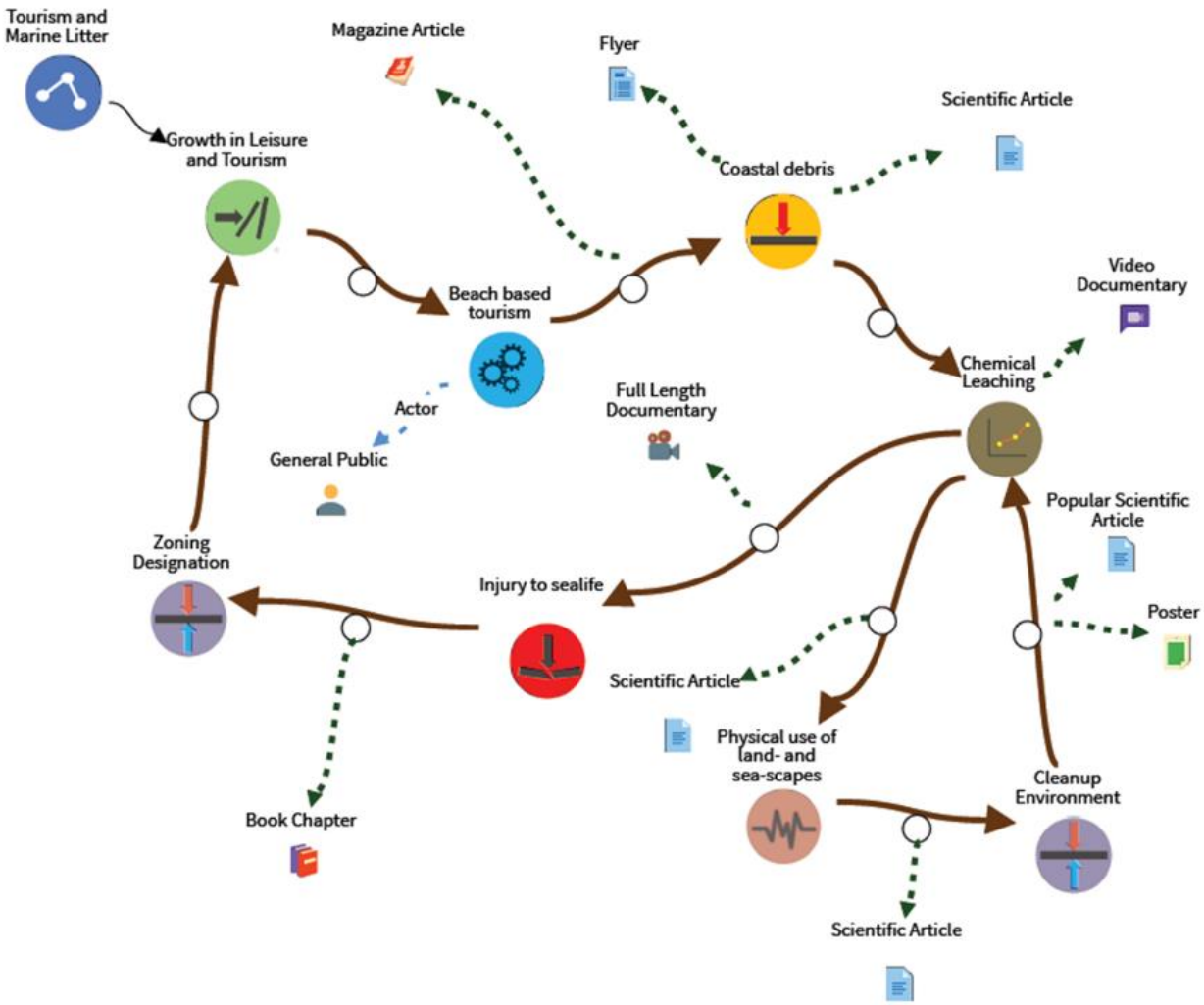



*Figure 3-1: A simple DAPSIWR Story*

Using the bulk Knowledge Upload facility, items of Knowledge can be attached to the nodes and links in the Story graph, providing a searchable knowledge resource for educators, journalists, policymakers Ocean Literacy professionals and the general public.

The ResponSEAble Knowledge Base was developed by NUIG between 2016 and 2019. It was built using the Graph Database (OrientDB), Java Enterprise Edition, RESTful web services, Rappid front-end (Javascript, HTML, jQuery etc) and is optimised for use with the Chrome browser. It is available at http://responseable.nuigalway.ie:8080/kb/.

## *Components of the Knowledge Base System*

The ResponSEAble Knowledge Base was developed using the NetBeans IDE. The overall architecture of the Knowledge Base is shown in Figure 3-2. It is a Java Maven Web Application that connects with a remote OrientDB graph database via the Java Graph (Tinkerpop) API, and exposes a RESTful web service API to clients using a singleton EJB. Clients make AJAX calls to RESTful web services and receive data in JSON form.

### Server

Developer machine setup (2016):

- OrientDB Community Edition (version 2.2.3)
- Java (JRE) 8u74 (64 bit). OrientDB requires Java.
- Java JDK 8u74 (64 bit)
- NetBeans (EE) IDE 8.1.
- Java Tinkerpop API
- RESTful web services
- KB Editor optimisation requires MySQL db XXX

## Client

- Rappid
- jQuery
- Bootstrap

## Knowledge querying:

- ChartJS
- Export buttons, sort, search and pagination features courtesy of jQuery plugin, DataTables, and Datatables File export

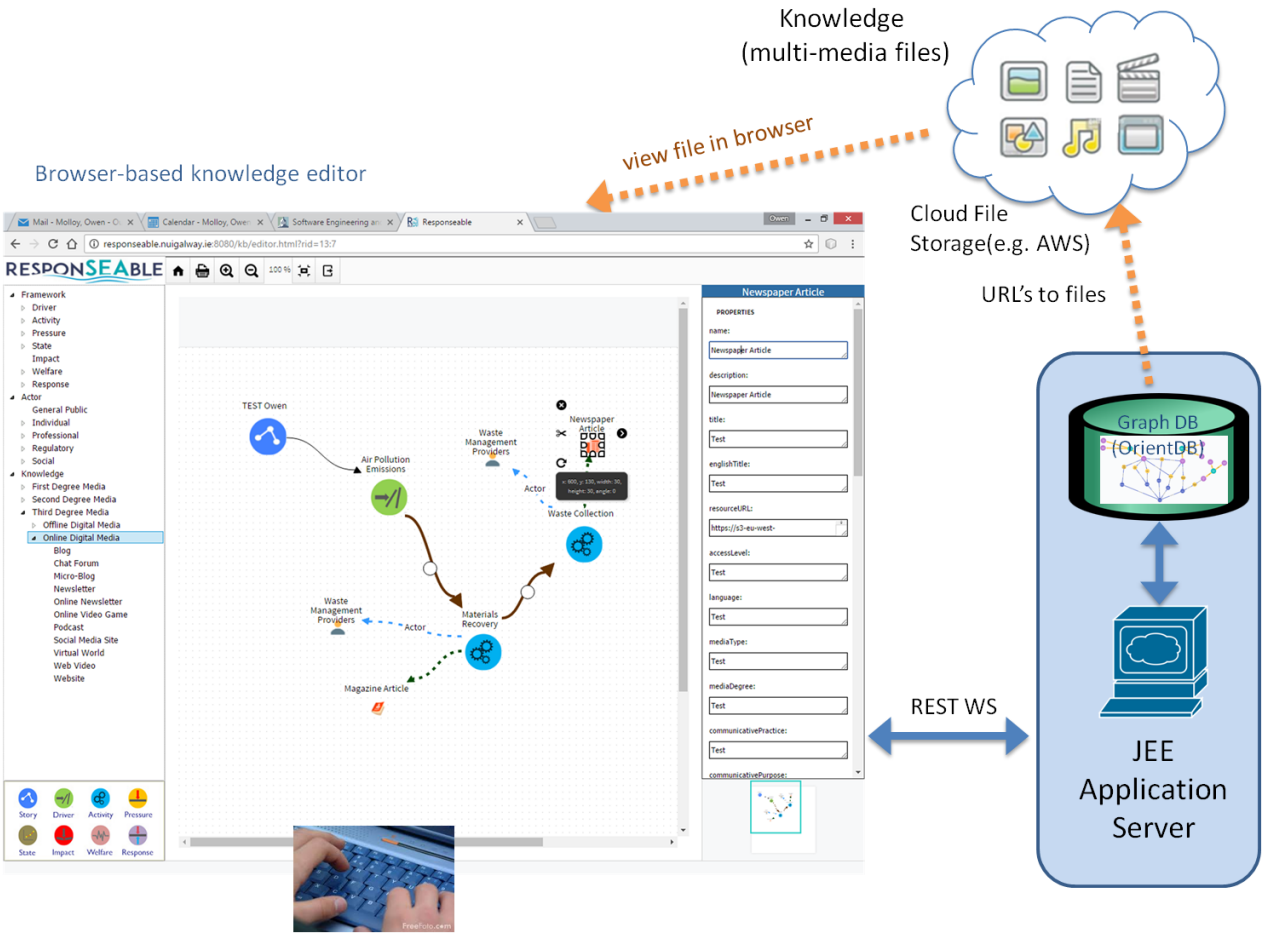



*Figure 3-2: Knowledge Base Architecture*

# 4. How to use the ResponSEAble Knowledge Base

## *Log in and home page*

The ResponSEAble Knowledge Base is available at http://responseable.nuigalway.ie:8080/kb.

To get a username and login password, you should contact the development team at: owen.molloy@nuigalway.ie.

Log in with your username and password. The home page is then displayed (see Figure 4-1).

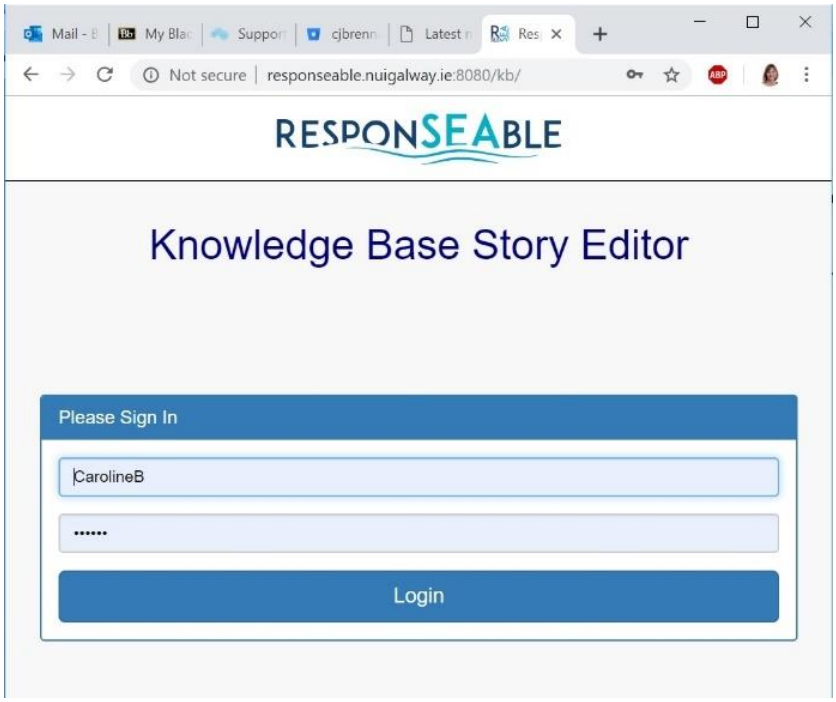



*Figure 4-1: Login Screen*

## *View or Edit an Existing Story*

Existing stories are listed on the users home page (see Figure 4-2). You will be able to see the Edit Story (pencil) icon under the Story name if you have edit access to it. Otherwise you will see only the View Story (eye) icon.

The Editor allows nodes and links to be added or changed. The Viewer looks very similar but does not allow changes. You can, however, click on a node in the Viewer to see its properties in the right-hand panel.

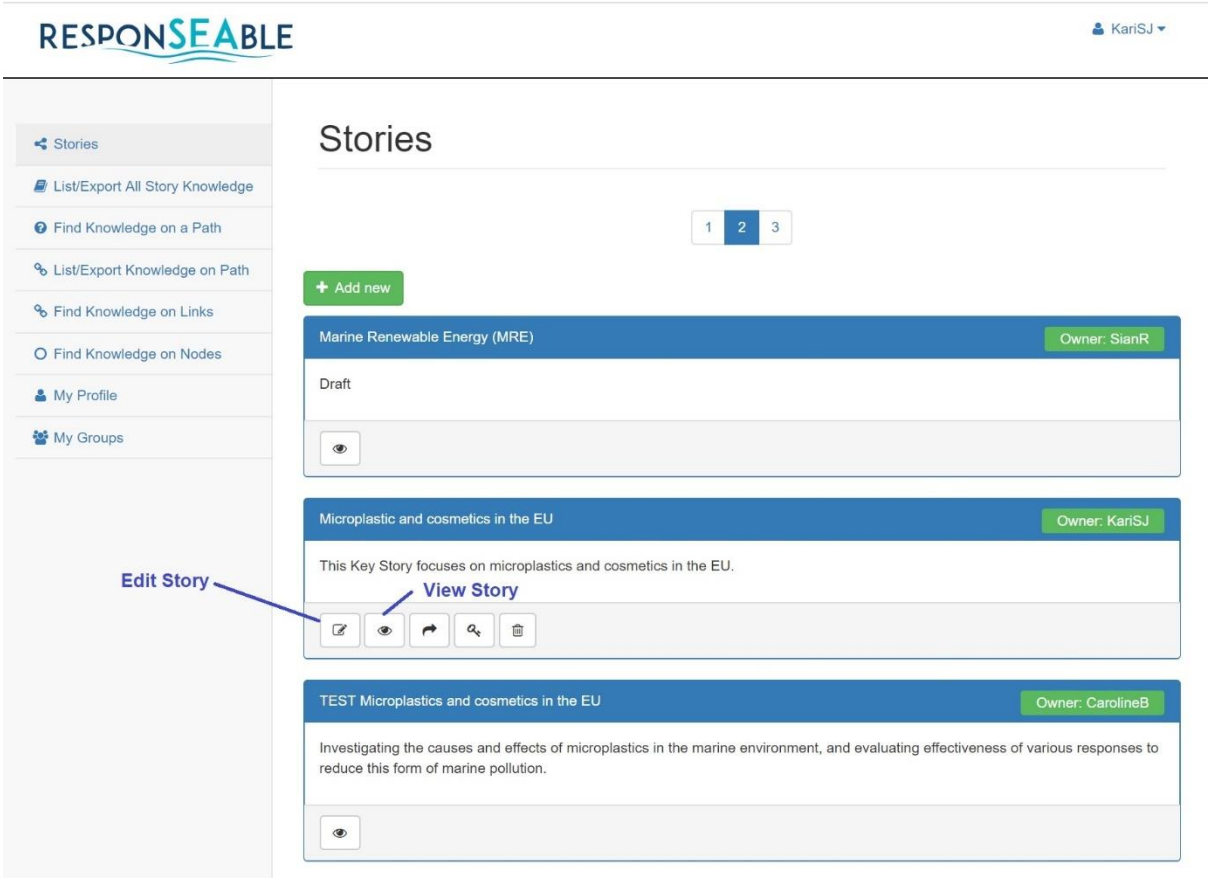



*Figure 4-2: User Home page, showing Edit and View Story icons*

## *Create a new DAPSIWR Story*

A story consists of a number of connected nodes, representing chains of cause-and-effect. Each of the nodes represents a DAPSIWR element.

To create a new DAPSIWR Story, follow these steps:

- Create the DAPSIWR story in the Editor, by adding DAPSIWR nodes and linking them
- Add Actors to nodes and links in the DAPSIWR story (also in the Editor)
- Add Knowledge to specific nodes and links in the DAPSIWR story, by importing a validated spreadsheet containing Knowledge details and

### Create the DAPSIWR in the Editor

From the home page, click the ‘Add New’ button and fill in basic information about the new Story (Figure 4-3).

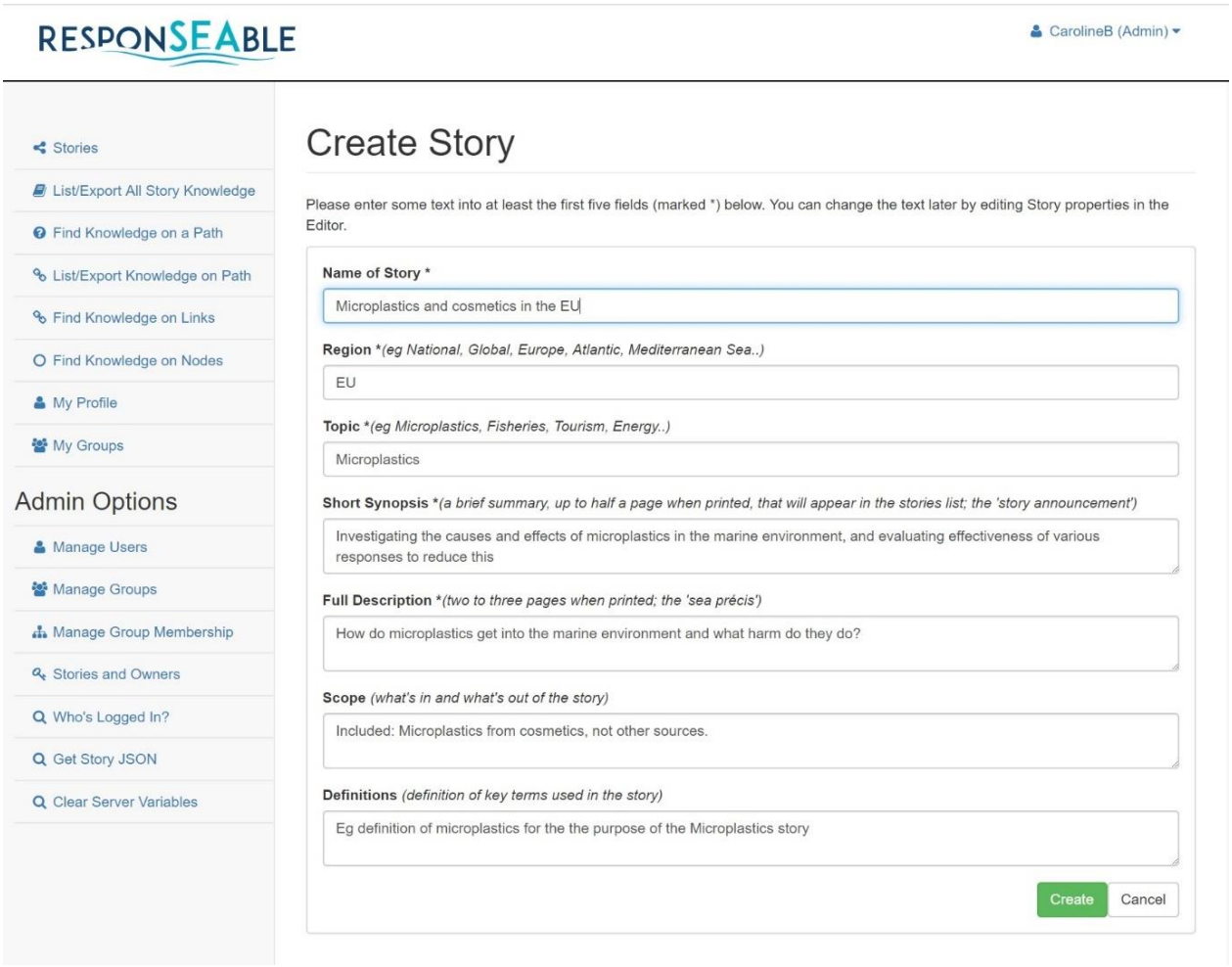



*Figure 4-3: Start by adding basic Story information*

Click the Create button. The Story just created will then be listed in the home page. Click the Edit Story button (pencil) icon to open the Story in the Editor (Figure 4-4).

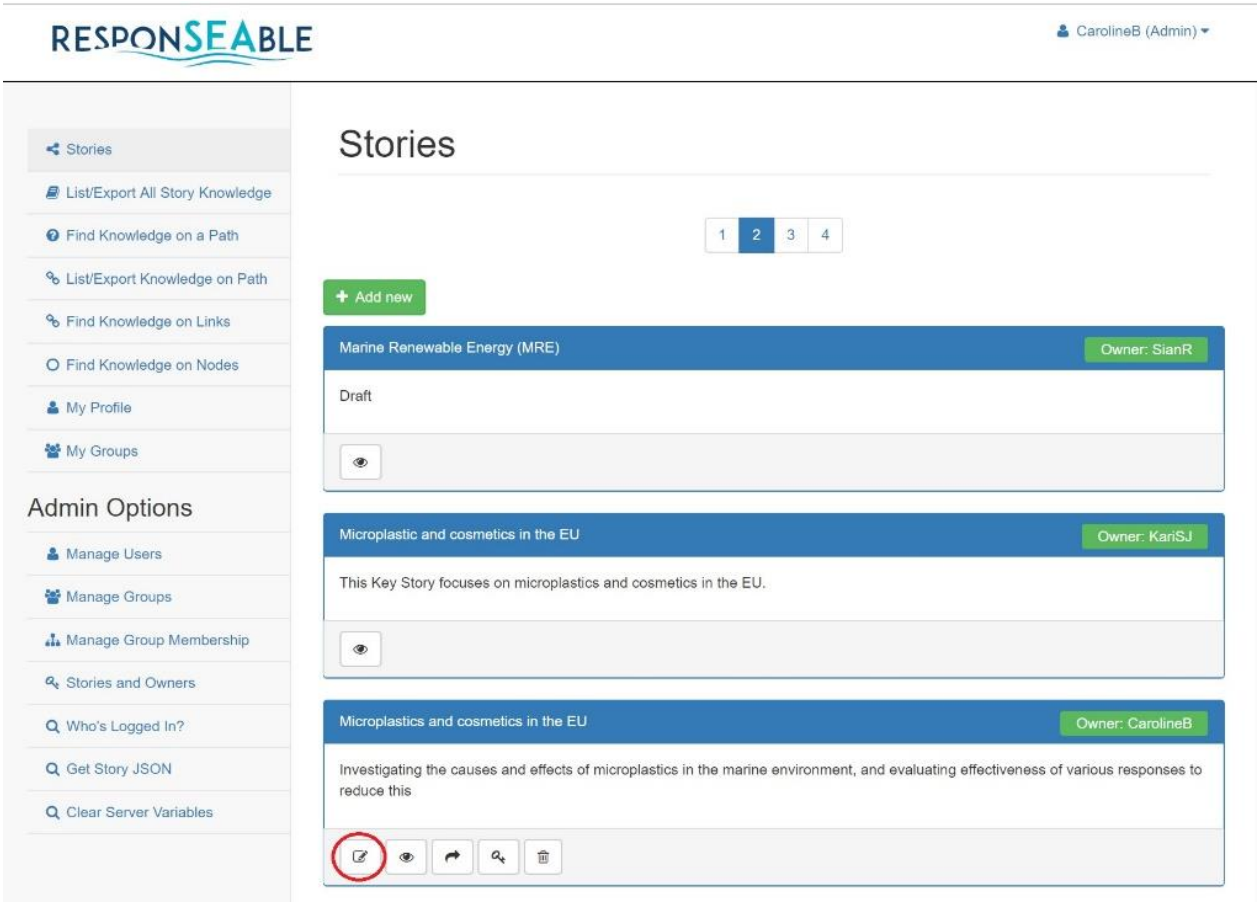



*Figure 4-4: Access the KB Editor to define the Story DAPSIWR using the Edit Story (pencil) icon*

A Story node (blue) will be seen on the canvas in the Editor, with the Story name shown above it. Its properties (shown in the right-hand panel) contain the general Story information you entered earlier. The DAPSIWR story must be built from that blue node, starting with Drivers.

Often you would add the nodes progressing in the DAPSIWR sequence – Drivers, then Activities, then Pressures, and so on. As you build the Story, you will have in mind specific Drivers, Activities, Pressures, and so on. You can select the closest specific DAPSIWR type using the tree-like hierarchy available under ‘Framework’ on the left-hand side of the Editor window.

For example, to select a particular Driver that is relevant to your story, open the ‘Framework’ classification hierarchy by clicking the small arrow to the left of the word ‘Framework’. Top level DAPSIWR categories then appear (Driver, Activity, Pressure, State, Impact, Welfare, Response). Open the Driver category to see more specific types of Driver. Explore available Drivers by clicking the small arrows to show or hide branches of the Driver hierarchy, until you find the most suitable category. If you are unsure of the meaning of a classification, you can hover the mouse over it (Figure 4-5).

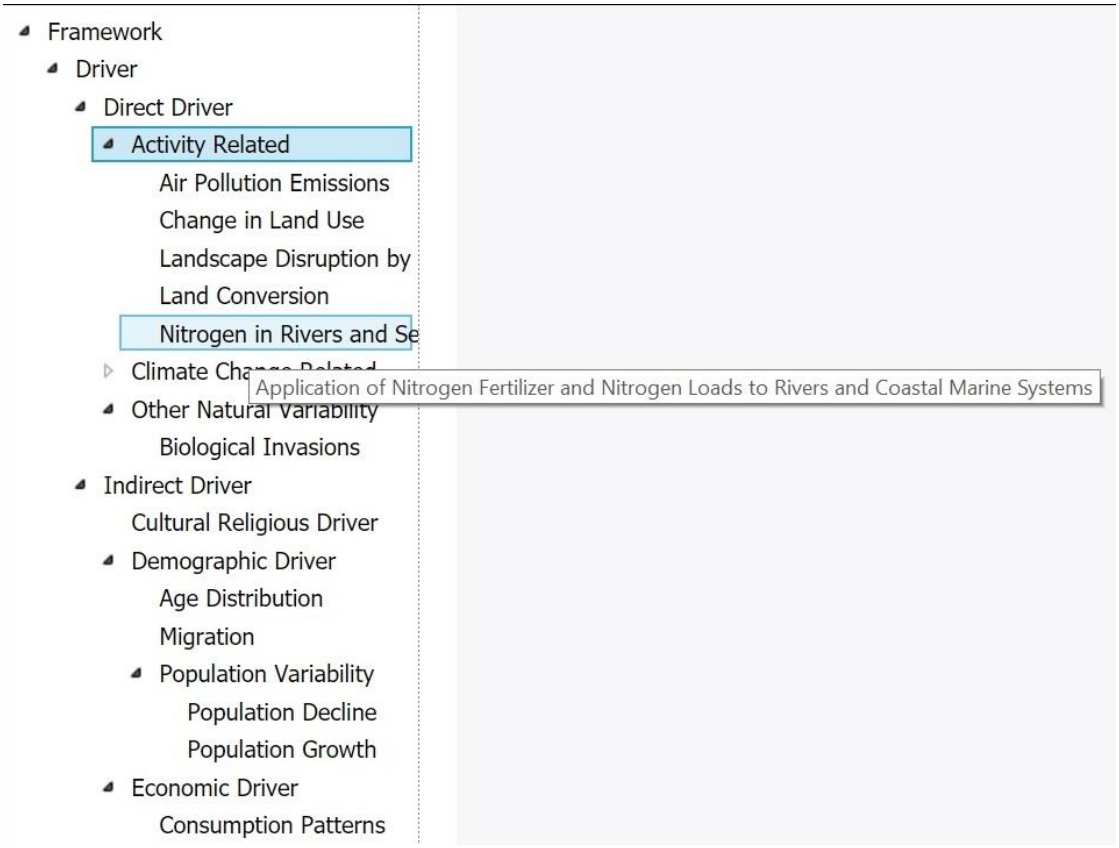



*Figure 4-5: KB Editor: Partly expanded Driver classification hierarchy with example tooltip*

When you have found the right classification, drag it over on to the canvas. A green circle (node) will appear there, denoting a Driver. See the legend for DAPSIWR categories used on the Editor canvas (Figure 4-6).


*Figure 4-6: Legend for DAPSIWR categories on the Editor canvas*

Add more information about this new Driver, by clicking it, and filling in the properties on the right- hand side of the editor window (Figure 4-7).

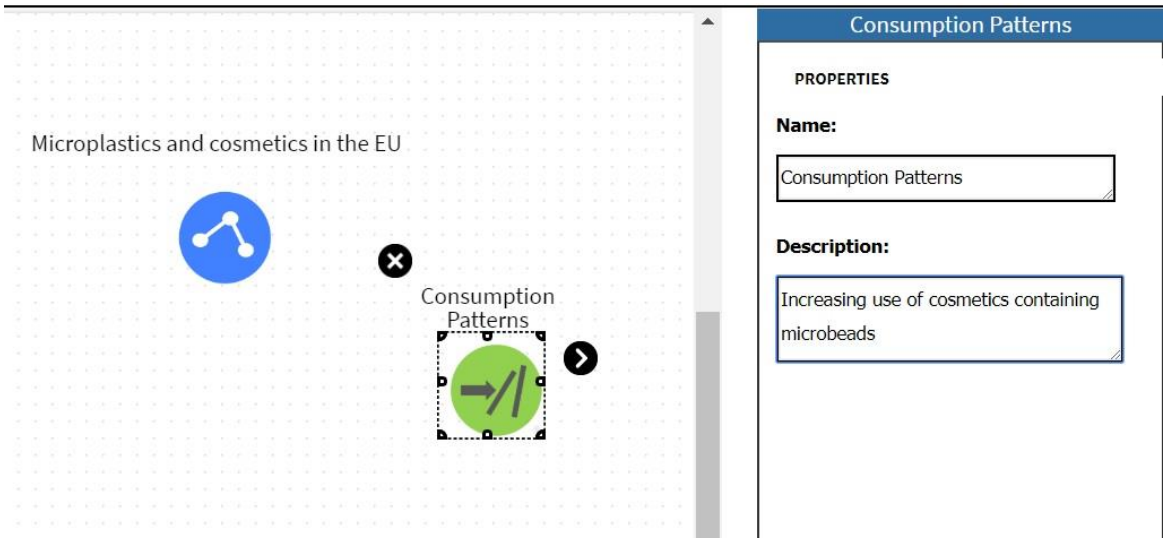



*Figure 4-7: Adding properties to the newly-added Driver*

Next, connect the new Driver to the Story. Click the Story node, and an arrow appears to the right. Click and drag it onto the Driver node, and release the mouse. An arrow now connects the two nodes (Figure 4-8).

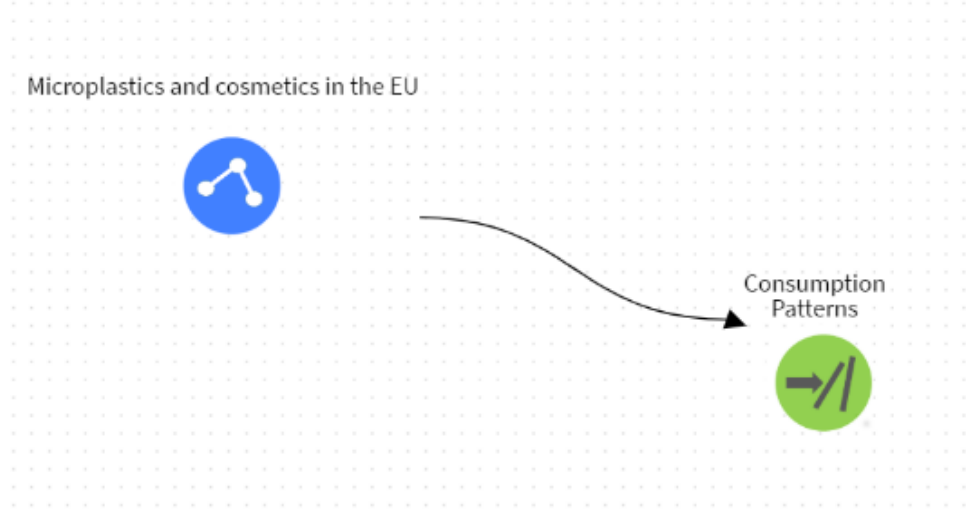



*Figure 4-8: The nodes are now linked*

You can add more Drivers, and connect them in a similar way.

Next, add Activities, add properties, and connect them. Continue with Pressures, States, Impacts, Welfare and Responses, until you have built a connected DAPSIWR story. The story is saved automatically as you work.

You can move nodes as you wish, in order to improve layout. We recommend positioning the Story node at the top, probably to the left, and the Story nodes arranged roughly left-to-right in DAPSIWR order. As you work, you might find the Zoom In, Zoom Out and Zoom to Fit icons useful.

If you need to delete a node, click it, and click the black X that appears beside it. Similarly, to delete a relationship between nodes, click it, and click the red X that appears beside it.

Continue working until you have defined your DAPSIWR Story. Here is simple Story, to illustrate (Figure 4-9).

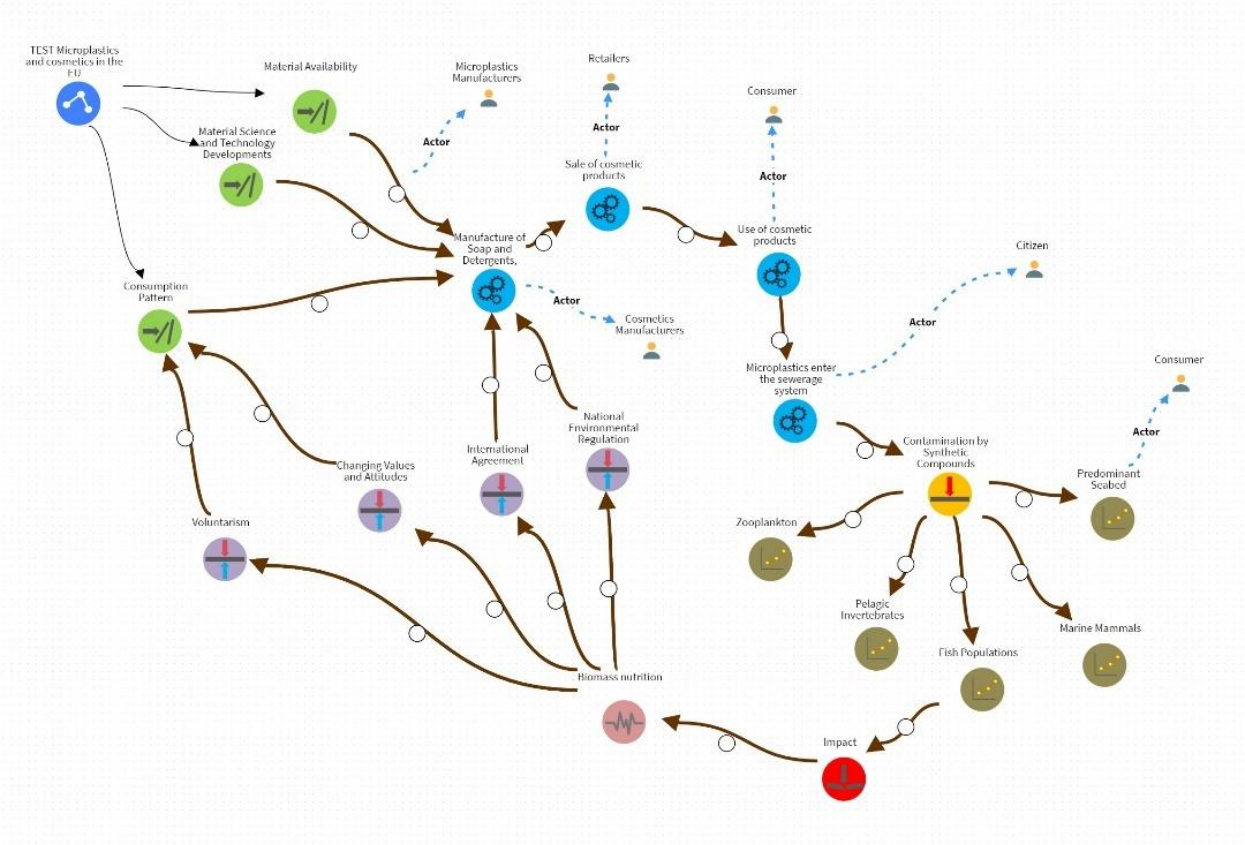



*Figure 4-9: Sample Story*

Notice that the arrows between DAPSIWR nodes represent causal relationships. They contain an intermediate node (a circle) which can be used to attach Actors or Knowledge to the relationships (see later sections).

Click the home icon to leave the Editor. 

## Add Actors to nodes and links in the DAPSIWR story

Actors – individuals, groups, companies, stakeholders – can be associated with DAPSIWR nodes, or the relationships between them. For example, Fishermen are associated with the Activity ‘Fishing’.

Return to the Editor. Open the ‘Actor’ classification hierarchy on the left-hand side (Figure 4-10).


*Figure 4-10: Actor classification hierarchy, partly revealed*

Proceed to add Actors to the Story in much the same way as you added the DAPSIWR nodes. That is:

- Find the appropriate Actor classification on the left-hand side
- Drag it over to the canvas
- Click it to add Actor properties (name and/or description)
- Connect it to the Story, by clicking the Actor node and dragging the arrow that appears beside it either to a node or a relationship between nodes (Figure 4-11). If connecting an Actor to a relationship/link, drag to the circle in the linking arrow.

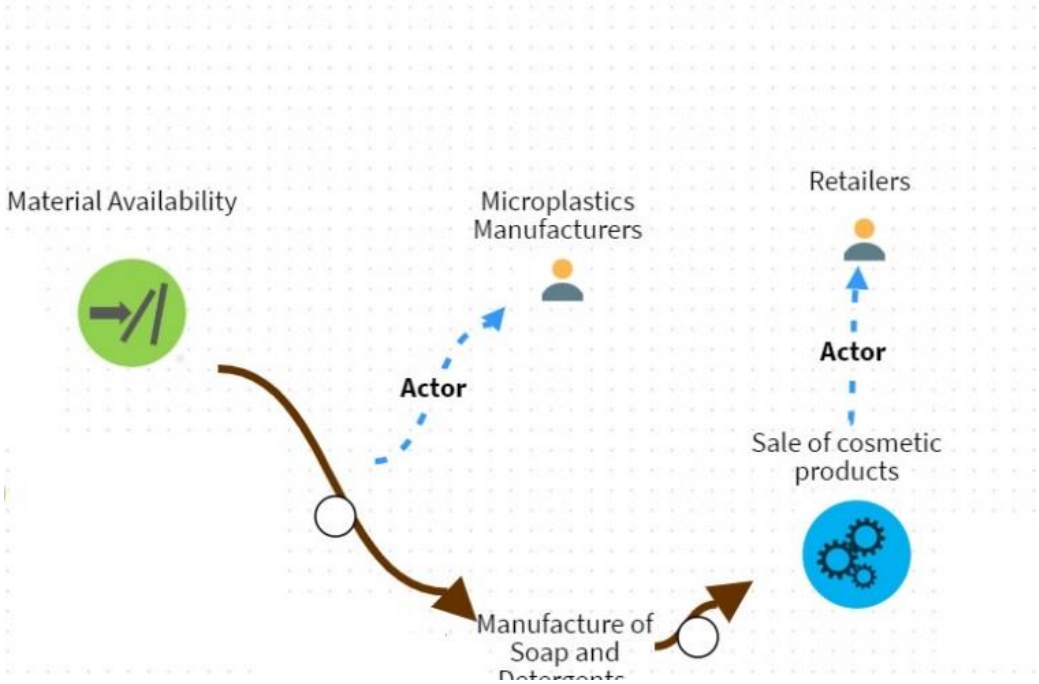



*Figure 4-11: One Actor linked to a node, another to a relationship*

## Add Knowledge to nodes and links in the DAPSIWR story

The term 'Knowledge' is interchangeable with 'Evidence'. Knowledge/Evidence may be a technical report, a website, a Youtube video, a poster etc. It illustrates or substantiates the aspect of the story it is connected to. Knowledge is usually associated with causal links (relationships between nodes), but sometimes is linked to nodes in the story, or the story itself, if very general.

### *Enter Knowledge details in a specially designed Excel spreadsheet*

Once the DAPSIWR Story has been created in the Editor, and Actors added to it, Knowledge (Evidence) can be added to any node or relationship in the Story. Whilst it is possible to attach it to the Story node where it is generally relevant, it is more useful to attach it to specific nodes and relationships to allow sophisticated Knowledge searches later. The DAPSIWR Story serves to structure the knowledgebase.

Adding Knowledge to the Story is not done using the Editor, but by populating a spreadsheet, which is then validated and imported. In this way, many Knowledge items can be added and described, and then imported into the Story in one operation.

The spreadsheet template is called *KB_evidence_2.1_template.xlsx*. It is set up to assist with adding and validating Knowledge properties, and includes helpful instructions and lists of available options.

Fill out one spreadsheet per Story, and include all Knowledge for the Story. In the spreadsheet:

- Fill in the Story name, together with the name, organisation and contact details of the person entering the data, in the 'Data Provider' sheet
- Complete the Knowledge ('Evidence') sheet (see example table below). This is the main section of the spreadsheet, and contains one row per item of Knowledge
- Use the 'Links' sheet to link the Knowledge to specific nodes or links in the Story DAPSIWR

Example of an entry in the 'Evidence' sheet (* denotes mandatory field):

| | | |
|---|---|---|
| ID* | A unique ID for this evidence | IAS-3 |
| Rating* | Rate the quality of this evidence in a range from 1 (poor quality) to 10 (highest quality) | 8 |
| Title* | Original title of the evidence | List of Invasive Alien Species of Union concern |
| English Title | Provide the English title, in case the original title is not in English | |
| Keywords* | One or more keywords related with this evidence, separated with comma | European Commission, invasive alien species, invasive species |
| Description | Description of the evidence | First list of invasive alien species published by the EU |
| Resource URL | In case there is a website related with this evidence please provide its URL | http://ec.europa.eu/environment/nature/invasivealien/list/index_en.htm |
| Level of Access* | Is the resource publicly accessible? | Public |
| Language* | Select the language used in the evidence | English |
| Main source | Name the main source (if applicable) | European Commission |
| Query | Search term that was used to obtain the data | |
| Order | Optional order of appearance (e.g. in Google search) | |
| Media Degree* | First, second or third degree media | third_degree |
| Media Group* | Online or offline digital media | online_digital_media |
| Media Type* | Choose from a list of media types (eg website, podcast, electronic flyer, ..) | website |
| Communicative Practice 1* | One-to-one, one-to-many, many-to-many | one to many |
| Communicative Practice 2 | Provide additional communicative practice if needed | |
| Communicative Purpose 1* | Choose from a list (eg inform, educate, advance knowledge, raise awareness, ..) | inform |
| Communicative Purpose 2 | Provide additional communicative purpose if needed | |
| Communicative Purpose 3 | Provide additional communicative purpose if needed | |

| | | |
|---|---|---|
| Target Group 1* | Provide the Target group, at least the first of three levels. List of Actors classification hierarchy provided | general_public |
| Target Group 2 | Provide additional target group if needed. | individual_actors: citizens |
| Target Group 3 | Provide additional target group if needed. | |
| Author | Author | EC |
| Publisher* | Name of publisher | EC |
| Organisational Sector* | List of Actors classification hierarchy provided | regulative_framework |
| Type of Organisation* | Provide the Type of Organisation, at least the first level. List of Actors classification hierarchy provided | legislators |
| Year* | Year of creation (4 digits) | 2017 |
| Time Reference | Period to which the evidence refers. From year - To year (4 digits each) | |
| Country of Origin | Mandatory if Geographical focus is "National" | |
| Geographical focus* | Geographical reference in the content: Global, Europe, Atlantic, Baltic Sea, Black Sea, Mediterranean Sea, North Sea | Europe |
| No of Hits | Hits of Website, Videostream | |
| No of Likes | Likes for social media content | |
| No of participants | Number of participants at any kind of gatherings | |
| No of retrievals | Number of downloads for any kind of downloadable content | |
| Impact Factor | Impact factor of scientific paper | |

Note: the above table refers to classification hierarchies for Actor and Media (Knowledge) types. These are listed in full in the Appendices of ResponSEAble Deliverable 3.1: 'Communication and Perception of the Human-Ocean Relationship'.

The next step is to provide information about where each item of Knowledge links in to the DAPSIWR Story – either to a node, or to a link between two nodes. This is done by filling in the 'Links' sheet (see Figure 4-12).

| * Evidence | From node | To node | * Relevance | |
|---|---|---|---|---|
| IAS-1 | Population Growth | Water Transport | 20 | |
| IAS-2 | Population Growth | Water Transport | 30 | |
| IAS-2 | Economic Distribution | Water Transport | 60 | |
| IAS-2 | Globalisation | Water Transport | 50 | |
| IAS-3 | Pelagic Components | XXX | 80 | |
| IAS-3 | Impact on pelagic components | | 80 | |
| | | | | |
| | | | | |
| | | | | |

Data Provider | Evidence | **Links**

*Figure 4-12: Example data in Links sheet*

The ID in the first column below must match exactly the ID of the Knowledge/Evidence in the 'Evidence' sheet. The node names must match exactly those found in the DAPSIWR editor. If the Knowledge is relevant to a node, fill in 'From node' and leave 'To node' blank. If it is relevant to a relationship, fill in both from and to nodes. How relevant? The figure in the Relevance column ranges from 0 (irrelevant) to 100 (absolutely relevant). Finally, some Knowledge is of general relevance to a Story. To attach Knowledge to the Story as a whole, enter 'Story' under 'From node', and leave 'To node' blank.

### *Validate the spreadsheet and then upload it using an online utility*

The application for submitting Knowledge spreadsheets is found here: http://78.47.243.169/import_evidence/

Two basic functionalities have been developed and can be executed from this page (Figure 4-13).

- File upload and validation of the data and
- Data submission and storage in the Knowledge Base

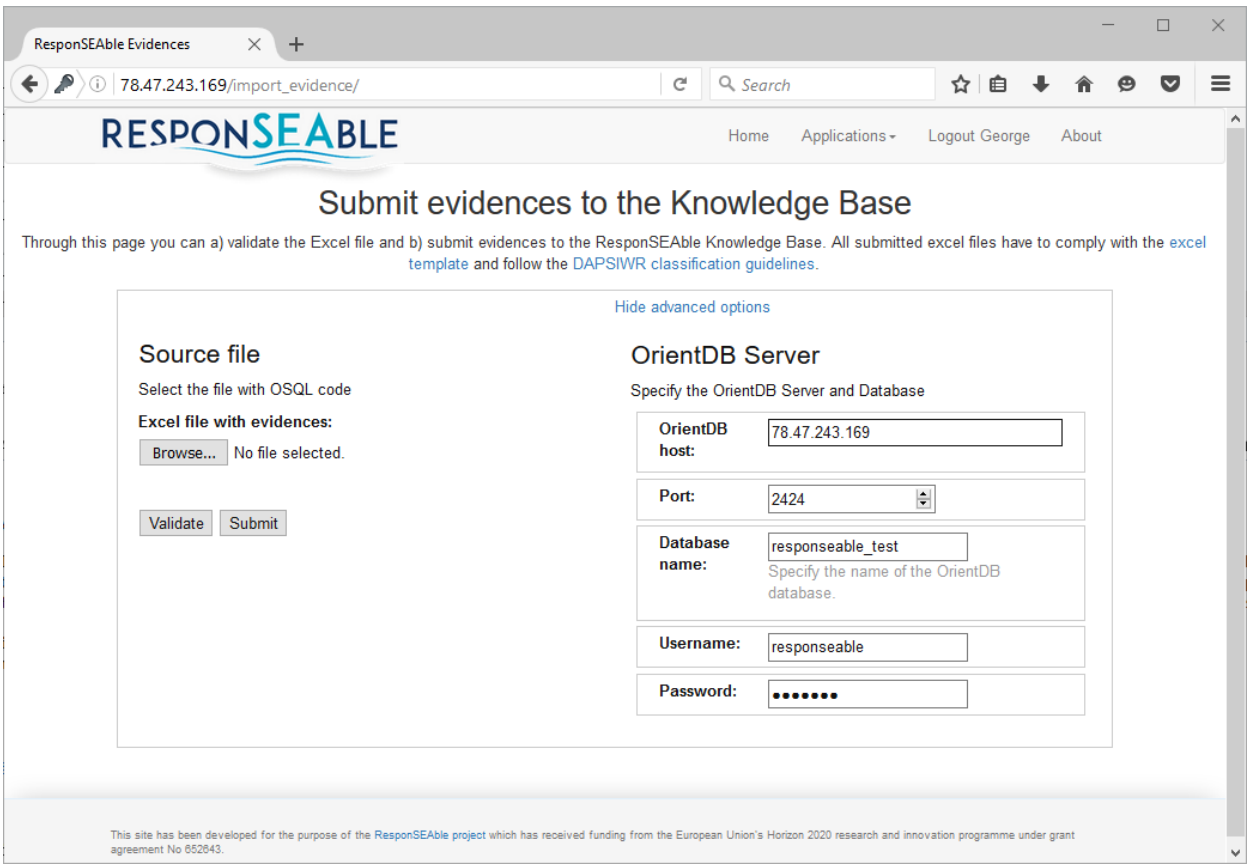
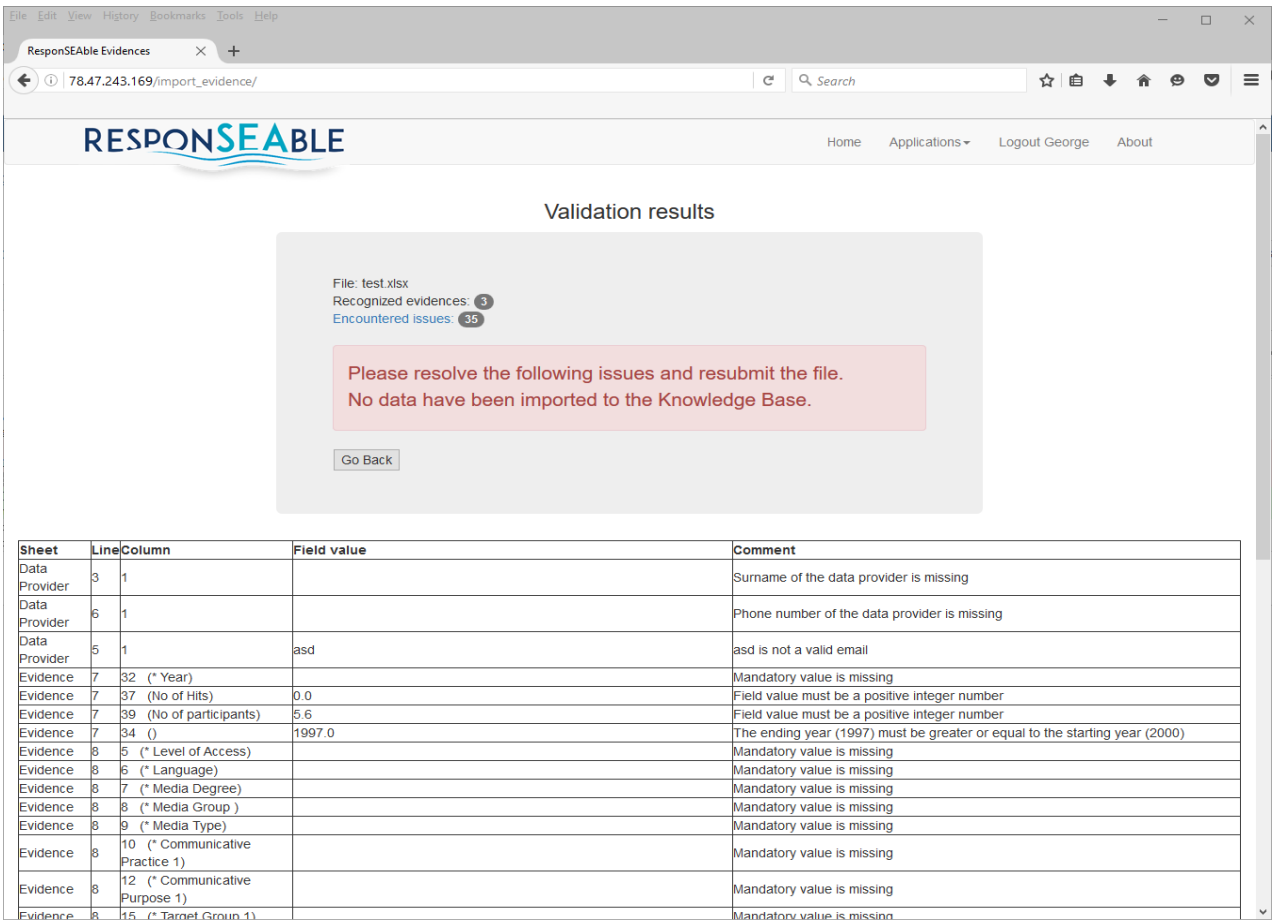



*Figure 4-13: Web application for submission of evidence to the Knowledge Base*

The parameters related to the OrientDB Server and Database can be specified through the respective fields under the *advanced options*, but default parameter values are given so that normally the user will not be concerned about this issue.

Browse to your spreadsheet, select it, and then click Validate. Checks will be performed on mandatory fields, field data types and whether duplicate Knowledge already exists in the database. If issues are encountered, the user is informed in detail and asked to correct the errors and resubmit the file (Figure 4-14).




| Sheet | Line | Column | Field value | Comment |
|---|---|---|---|---|
| Data Provider | 3 | 1 | | Surname of the data provider is missing |
| Data Provider | 6 | 1 | | Phone number of the data provider is missing |
| Data Provider | 5 | 1 | asd | asd is not a valid email |
| Evidence | 7 | 32 (* Year) | | Mandatory value is missing |
| Evidence | 7 | 37 (No of Hits) | 0.0 | Field value must be a positive integer number |
| Evidence | 7 | 39 (No of participants) | 5.6 | Field value must be a positive integer number |
| Evidence | 7 | 34 () | 1997.0 | The ending year (1997) must be greater or equal to the starting year (2000) |
| Evidence | 8 | 5 (* Level of Access) | | Mandatory value is missing |
| Evidence | 8 | 6 (* Language) | | Mandatory value is missing |
| Evidence | 8 | 7 (* Media Degree) | | Mandatory value is missing |
| Evidence | 8 | 8 (* Media Group ) | | Mandatory value is missing |
| Evidence | 8 | 9 (* Media Type) | | Mandatory value is missing |
| Evidence | 8 | 10 (* Communicative Practice 1) | | Mandatory value is missing |
| Evidence | 8 | 12 (* Communicative Purpose 1) | | Mandatory value is missing |
| Evidence | 8 | 15 (* Target Group 1) | | Mandatory value is missing |

*Figure 4-14: Validation results page*

Once corrected, click the Submit button. The file will be validated again, and if it passes, the data will be imported into the specified database.

Note that if you later decide to make changes to your spreadsheet, such as adding new Knowledge, removing Knowledge, or changing Knowledge details, when you revalidate and resubmit the modified spreadsheet, all previous Knowledge will be replaced with the Knowledge contained in the latest spreadsheet.

# 5. Knowledge Querying

In a DAPSIWR Story, Knowledge can be linked to an individual DAPSIWR node, or, more commonly, to the causal link between two nodes.

As previously described, this 'Knowledge' takes a variety of forms, such as technical reports, websites, scientific articles, books, magazine articles, films, exhibitions, performances and so on. Its function can be *to illustrate* a causal link, and/or to support or *provide evidence* for it. An example of Knowledge illustrating a link might be a Youtube video to show the connection between sewerage outflow and the resulting effect on sea life. An example of Knowledge providing evidence for a link might be a detailed scientific report.

The Knowledge Base offers sophisticated Knowledge querying based not only on traditional text or property (field) searches, but on *relevance to particular nodes, links, or sections* of the DAPSIWR Story. So, for example, if an educator is interested in finding material to illustrate or provide evidence for a link between a specific Activity and a specific Impact, it is easy to find that link and any Knowledge associated with it in the database. In this way, the expertise of the creator of the DAPSIWR Story in the Knowledge Base, in matching specific Knowledge to specific parts of the DAPSIWR, is made available to general users. This potentially gives them much more meaningful and targeted results than, say, a Google search on the subject.

The following sections will illustrate the Knowledge querying facilities using the Invasive Alien Species (IAS) Story (Figure 5-1).


*Figure 5-1: Sample Story*

The exchange of ballast water and release of hull fouling (Figure 5-2) in a marine environment are today the main activities linked to the introduction of IAS and a leading cause of biodiversity loss in Europe and across the globe. It can cause extensive ecological and economic damage to aquatic ecosystems and may be responsible for serious human health issues.

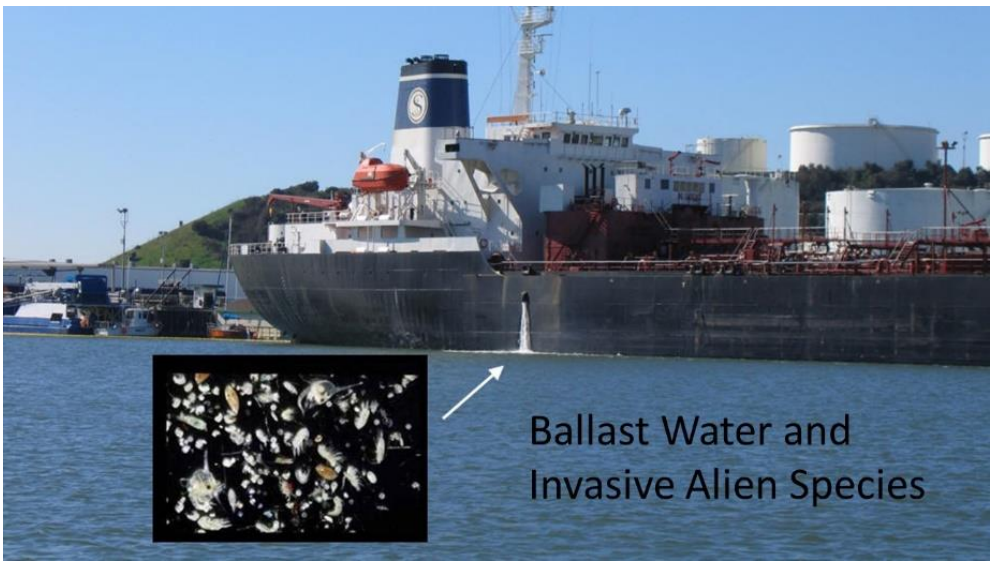



*Figure 5-2: Ballast Water Release*

Taking the example of the Invasive Alien Species Story, the ResponSEAble Knowledge Base queries are designed to answer questions such as these:

- What Knowledge is in the Knowledge Base about the whole IAS DAPSIWR story?
- What Knowledge / Evidence is there about the link between two nodes on a path in the DAPSIWR (not necessarily adjacent)? For example, between Water Transport (an Activity) and Alteration of Ecosystem Functioning (an Impact)?
- How good is that Knowledge, and how relevant is it?
- What links between non-adjacent nodes Water Transport (ACTIVITY) and Alteration of Ecosystem Functioning (IMPACT) are supported with Knowledge / Evidence? And which are not?
- Where can I find a particular item of Knowledge online, and what are all the details, such as target audience, geographical focus, quality, author, publication?
- What nodes in the Story are illustrated with Knowledge / Evidence? Are there any videos, websites, reports?

There are five menu options for Knowledge Querying, shown in yellow in Figure 5-3. These will be described in turn in the following pages.

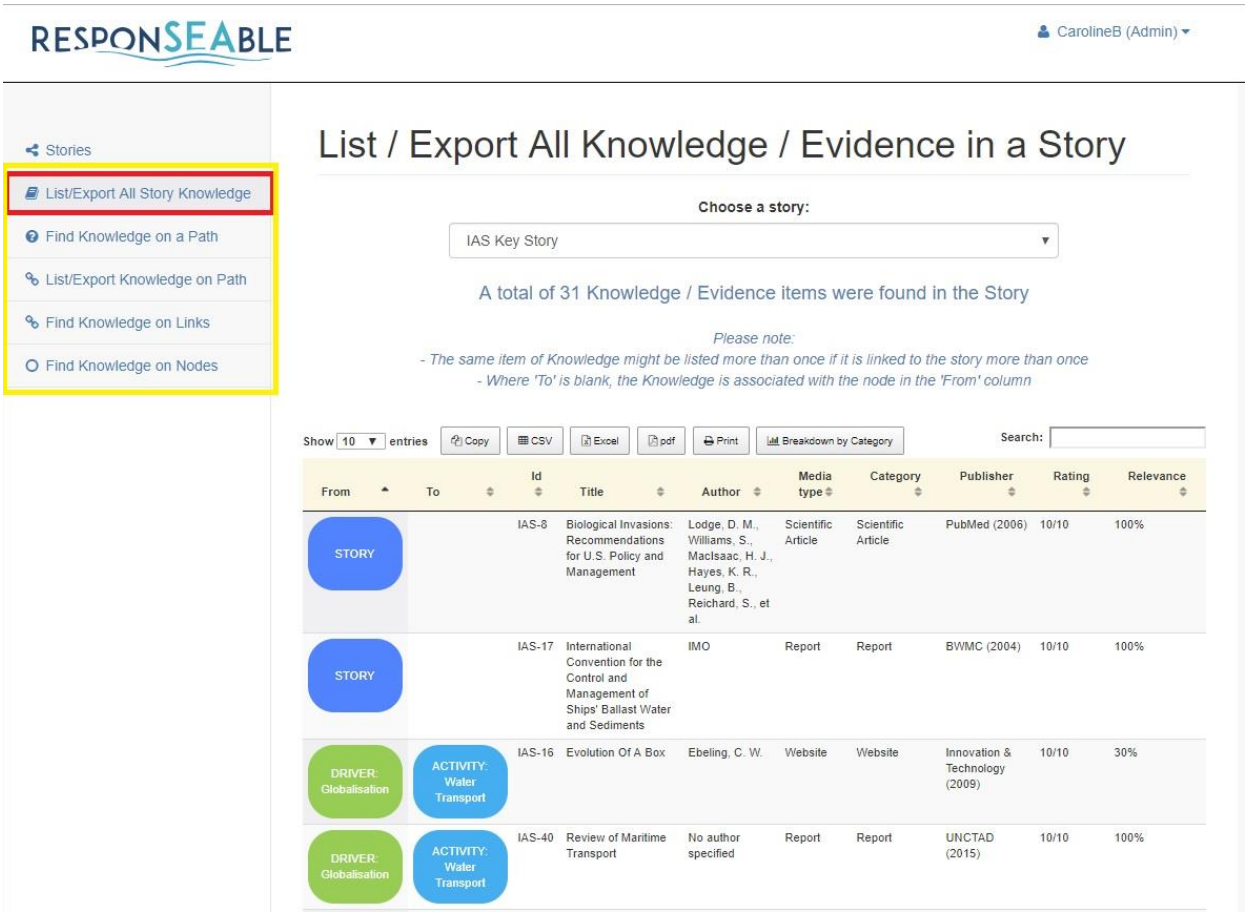



*Figure 5-3: Knowledge query options, with first option shown*

## *List / Export All Story Knowledge*

The first option is the most general. It retrieves a list summarizing all Knowledge for a specific story. Each row represents one item of Knowledge. If the Knowledge is linked to a specific node, it is shown in the first column, and the second column is blank. If nodes are shown in both columns, then the Knowledge is linked to the relationship between them (Figure 5-3).

You can search, sort, export, print and chart the Knowledge in various ways, as outlined below.

| | |
|---|---|
| **To sort by a column** | **Click the up arrow in the column heading to sort in ascending order, or the down arrow to sort in descending order** |
| **To view all the details of an item of Knowledge** | Click on any row. All details will be shown, including a link to the online resource in most cases (see Figure 5-4). |
| **Search** | Enter text in the Search box, e.g. 'ballast', to filter the rows. To clear the filter, simply remove the search text. |
| **Find Knowledge of a particular type** | To find all websites, for example: you can enter 'website' in the Search box, and/or sort by Media type or Category columns and scroll up or down the list |

| | |
|---|---|
| **Copy** | Copies rows currently displayed to the clipboard – paste elsewhere as necessary |
| **Export rows** | Click the CSV, Excel or pdf buttons to export rows currently displayed (current search filter will affect results) |
| **Print** | Click the Print button to print currently displayed rows from your browser (current search filter will affect results) |
| **Display a chart of Knowledge broken down by Category** | Click the 'Breakdown by Category' button (current search filter will affect results). See Figure 5-5. |

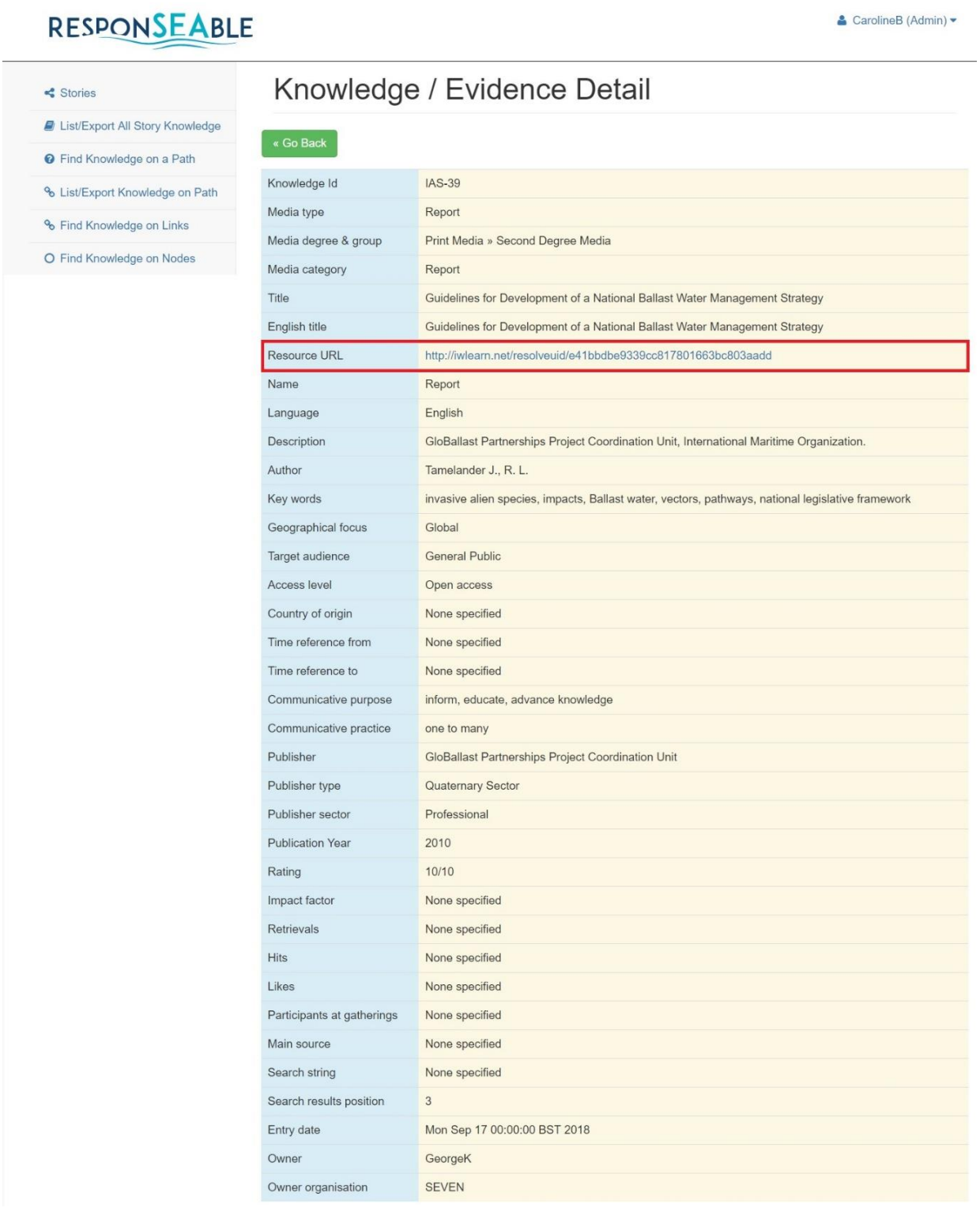



*Figure 5-4: Click a Knowledge row to get all the details, including a link, if available (highlighted)*

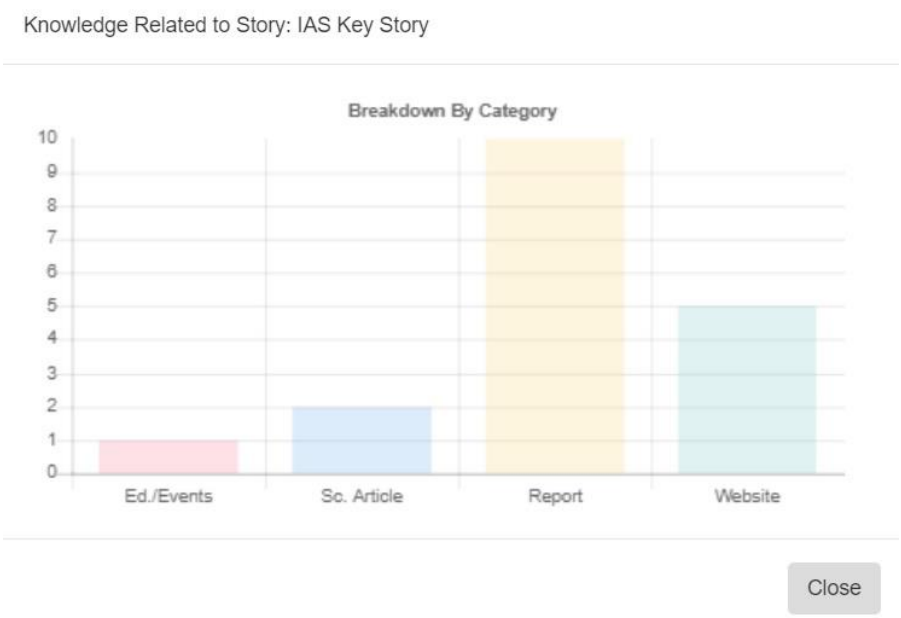



*Figure 5-5: Chart showing breakdown of Story Knowledge by Media Category*

## *Find Knowledge on a Path*

This option is useful for finding what Knowledge or Evidence exists to support a path in the DAPSIWR Story. The nodes can be adjacent or not, but they must be connected. In the example shown in Figure 5-6, we ask, what do we know about the link between Water Transport (an Activity) and the Alteration of ecosystem functioning (an Impact)?

Based on the Knowledge entered into the database, the answer is that the causal chain, or path, includes four nodes (Activity – Pressure – State – Impact), with three relationships joining them, and that 3 items of Knowledge exist on the first link, and none on the others. This option presents the answer in a link-orientated way. It is useful for showing where Knowledge exists, and where the gaps are. The results are based on the shortest path between two nodes.

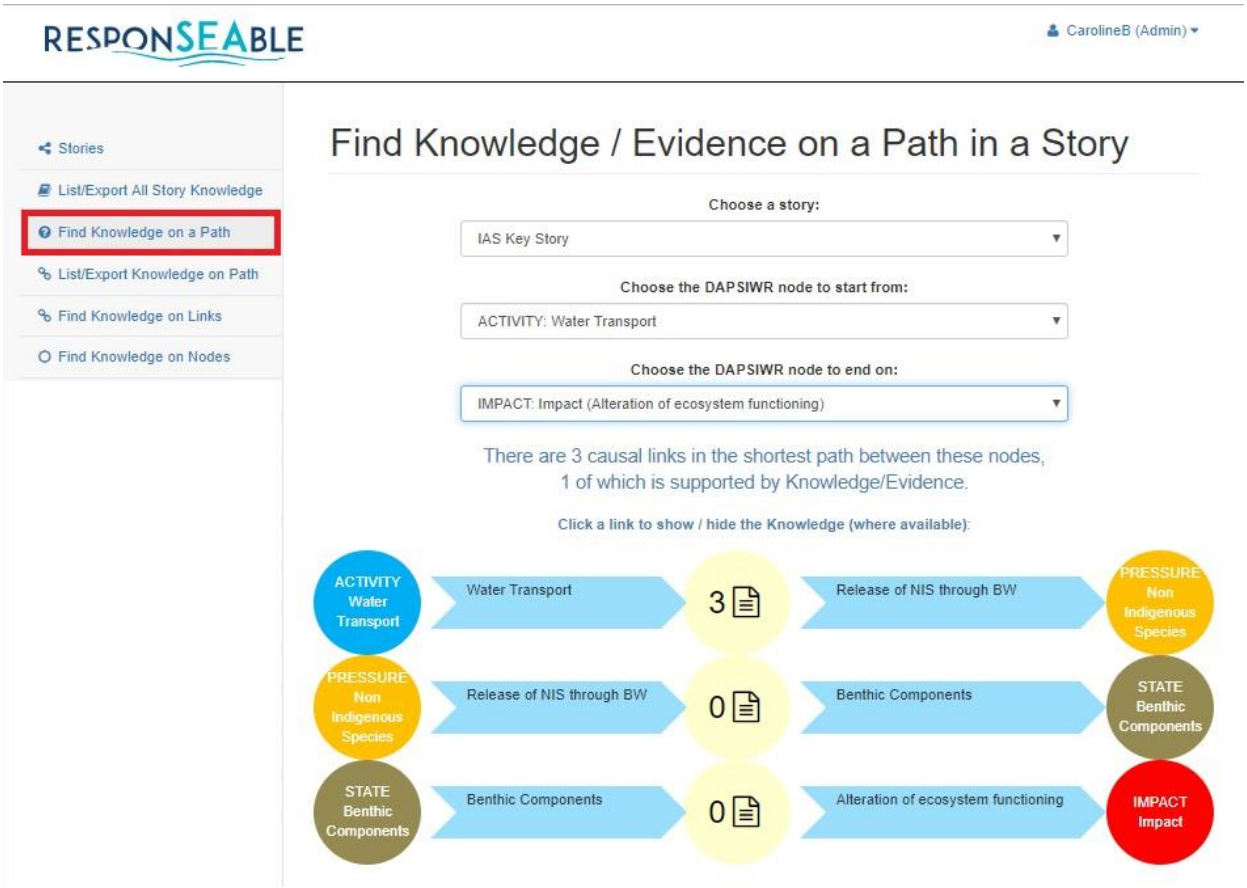
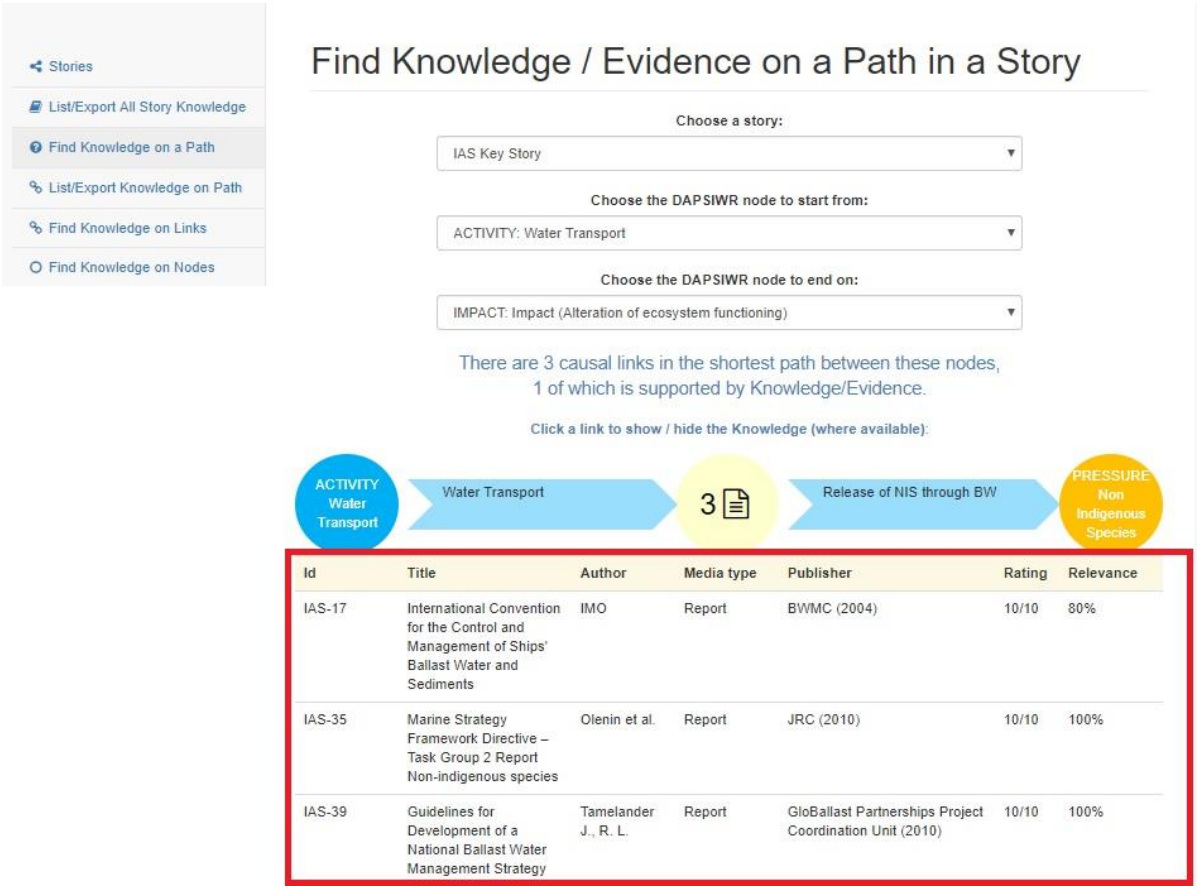



*Figure 5-6: Second query option*

You can click on a link that has Knowledge on it to open a summary list of that Knowledge (see Figure 5-7). From there, as usual, you can click on any row to see all the details of that Knowledge item, including a live link to the online resource, if available. Or click the row again to close the list.




| Id | Title | Author | Media type | Publisher | Rating | Relevance |
|---|---|---|---|---|---|---|
| IAS-17 | International Convention for the Control and Management of Ships' Ballast Water and Sediments | IMO | Report | BWMC (2004) | 10/10 | 80% |
| IAS-35 | Marine Strategy Framework Directive – Task Group 2 Report Non-indigenous species | Olenin et al. | Report | JRC (2010) | 10/10 | 100% |
| IAS-39 | Guidelines for Development of a National Ballast Water Management Strategy | Tamelander J., R. L. | Report | GloBallast Partnerships Project Coordination Unit (2010) | 10/10 | 100% |

*Figure 5-7: Click a link row to see the Knowledge associated with the link*

## *List/Export Knowledge on a Path*

This option is similar to the last option, but instead of presenting the answer that emphasises the links, and groups Knowledge on those links, it presents a list of all the Knowledge on the path between the two nodes (Figure 5-8). Link information is still shown, in the first two columns.

You can sort, search, export, print or chart the results, using the same techniques as you did in the first search option.

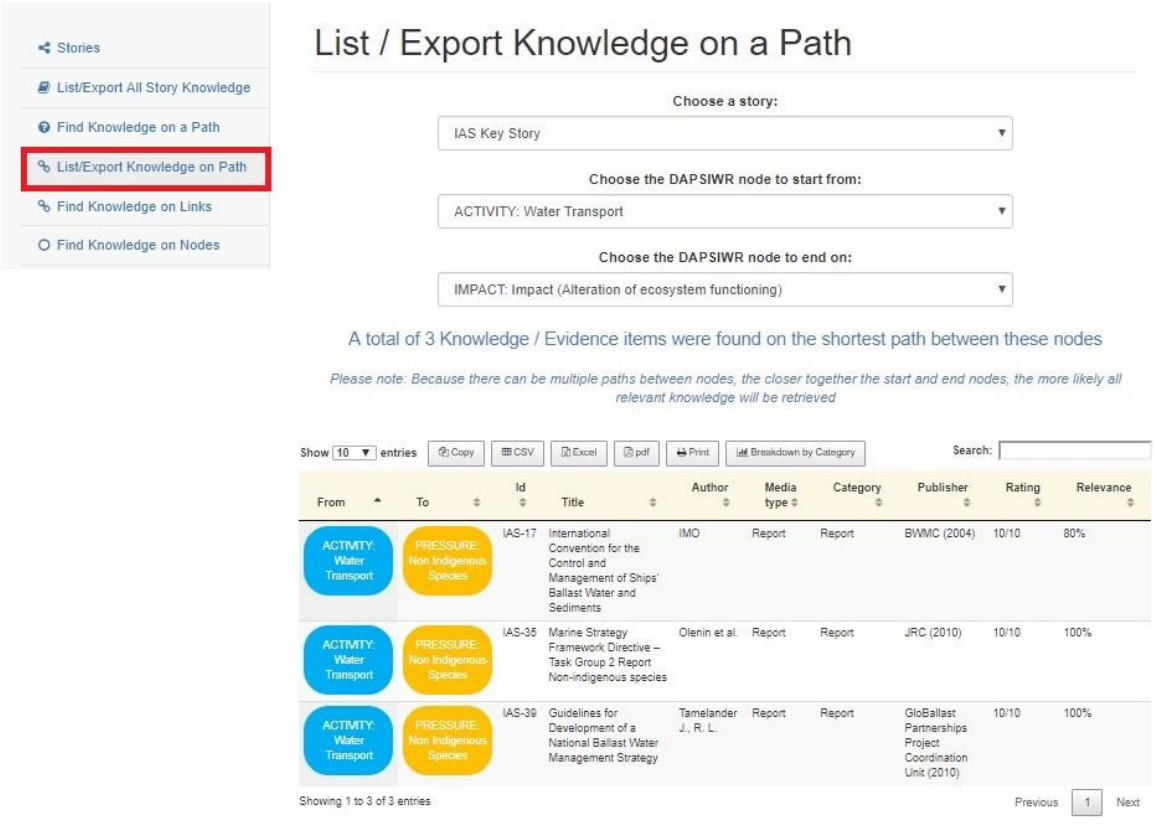



*Figure 5-8: Third search option*

## *Find Knowledge on Links*

This option is useful for listing all relationships/links in a Story that have supporting Knowledge. It excludes links in the Story that don't have supporting Knowledge. Since the Knowledge is associated with links, in this context it will serve as evidence for those links, or for illustrating them.

Once again, click any link row to see the Knowledge associated with the link (see Figure 5-9), and click any of the Knowledge rows to get the full details, including a live link to the online resource, if available.

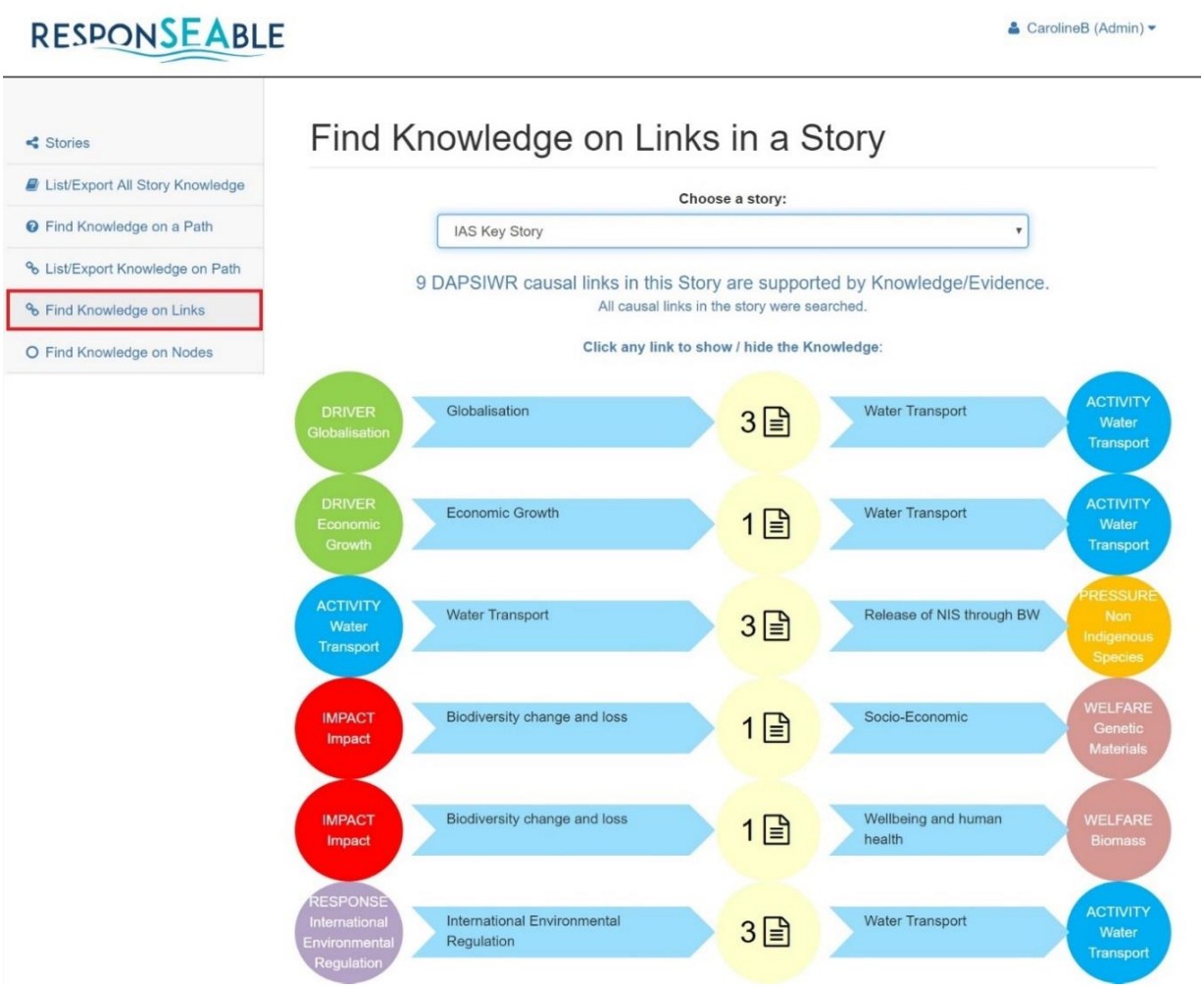



*Figure 5-9: Fourth search option*

## *Find Knowledge on Nodes*

This option is useful for listing all DAPSIWR nodes in a Story that have supporting Knowledge. In this context, the Knowledge will probably serve to illustrate a node, e.g. an Activity such as Water Transport.

Note that Knowledge associated with the Story link is of general relevance to the Story. An example would be an introductory Youtube video that gives a general overview of the problem of Ballast Water and Alien Invasive Species.

Once again, click a row to see the Knowledge associated with a node, and click any of the Knowledge rows to get the full details, including a live link to the online resource, if available (Figure 5-10).

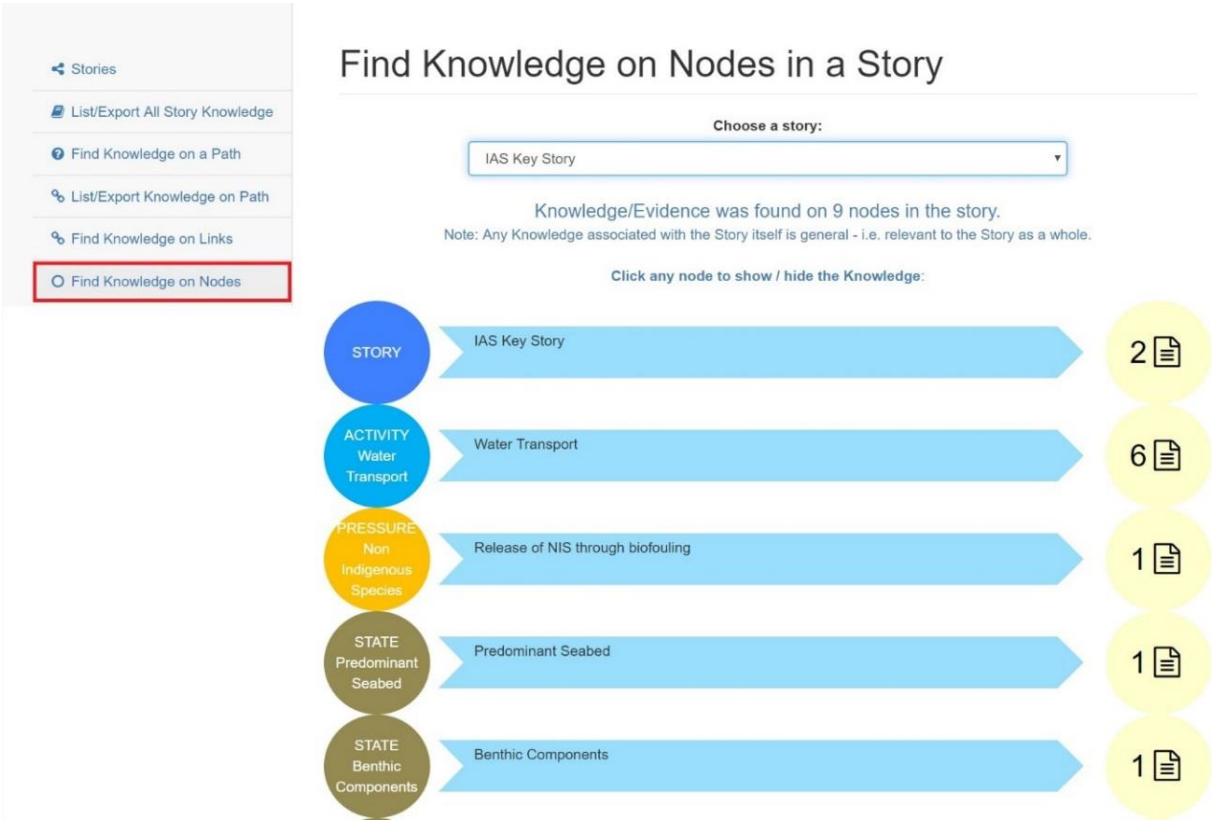



*Figure 5-10: Fifth search option*

# 6. Browser-based Knowledge Base Viewer

The KB Viewer is hosted at http://responseable.nuigalway.ie/kbviewer. It is a browser-based tool, so it is not necessary to install any software on the client machine other than a web browser. The best browser with which to view the tool is Chrome (https://www.google.com/chrome/) as it full supports the software needed to run. The tool is built using the Cytoscape.js open source graph theory library[2], the expand-collapse extension[3], jQuery[4], Bootstrap[5] and Chart.js[6].

When the tool opens the user must use the 'SELECT THE STORY' drop-down list on the top right hand side (Figure 6-1) to select the story which they want to view.

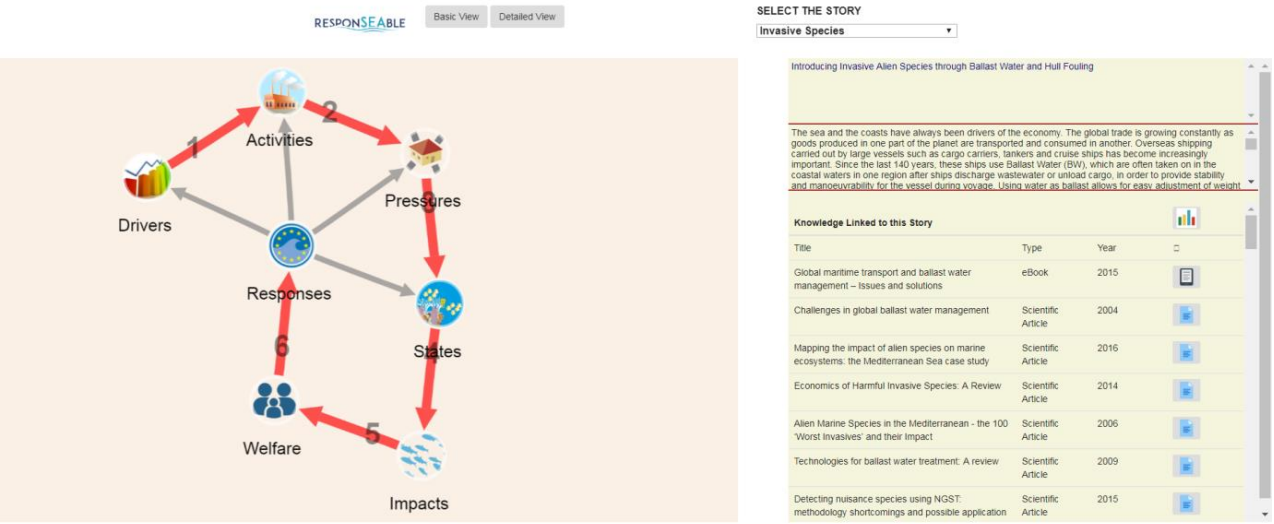



*Figure 6-1: The KB Viewer first screen*

## *Navigation in the Basic View*

When the user selects a story the high level DAPSIWR is display in the left hand pane, and on the right is displayed the story, and tables of the knowledge and actors which are to be found in the story. By clicking on the high level descriptor, such as Activities, it expands, and the user can see the actual Activities which are included in the Key Story (Figure 6-2). On the right hand side is then also displayed the knowledge and actors for all the Activities in the story. Clicking on a specific activity icon leads to the display of the detail (description, knowledge, actors) for that activity on the right hand side. Clicking on the icon beside any

[2] http://js.cytoscape.org

[3] https://github.com/iVis-at-Bilkent/cytoscape.js-expand-collapse

[4] http://www.jquery.com

[5] https://getbootstrap.com/

[6] http://chartjs.org/

knowledge item opens a pop-up window (Figure 6-3) containing the detailed information on the knowledge item, including the link to the resource if there is one.

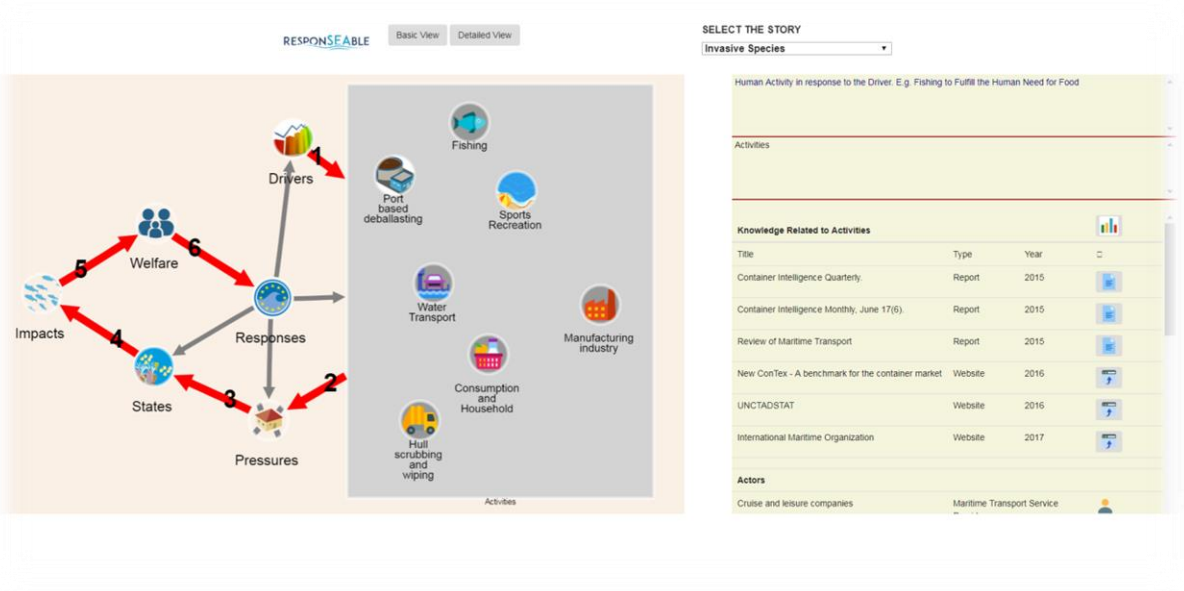



*Figure 6-2: Expand high-level node to see detail*

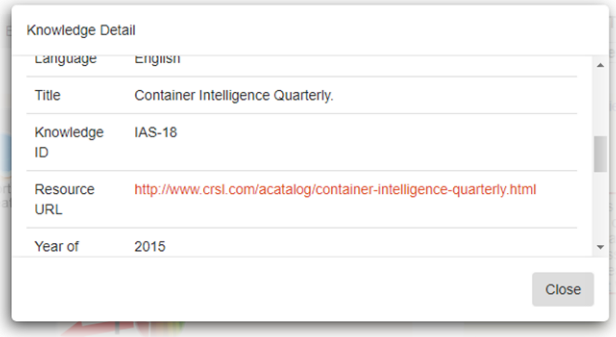



*Figure 6-3: Knowledge detail pop-up window*

If the user would like to see a chart summarising the knowledge at any level, for example story level as shown in , then they just click on the 'chart' icon above the knowledge table. The chart shows the number of knowledge items by category (Figure 6-4).


*Figure 6-4: Knowledge summary chart*

To see the detail relating to a specific node (for example 'Water Transport, as shown in Figure 6-5), the user clicks on that node. The node is highlighted in yellow, and the

knowledge and actors *directly* linked to that node are displayed in the table on the right hand side.

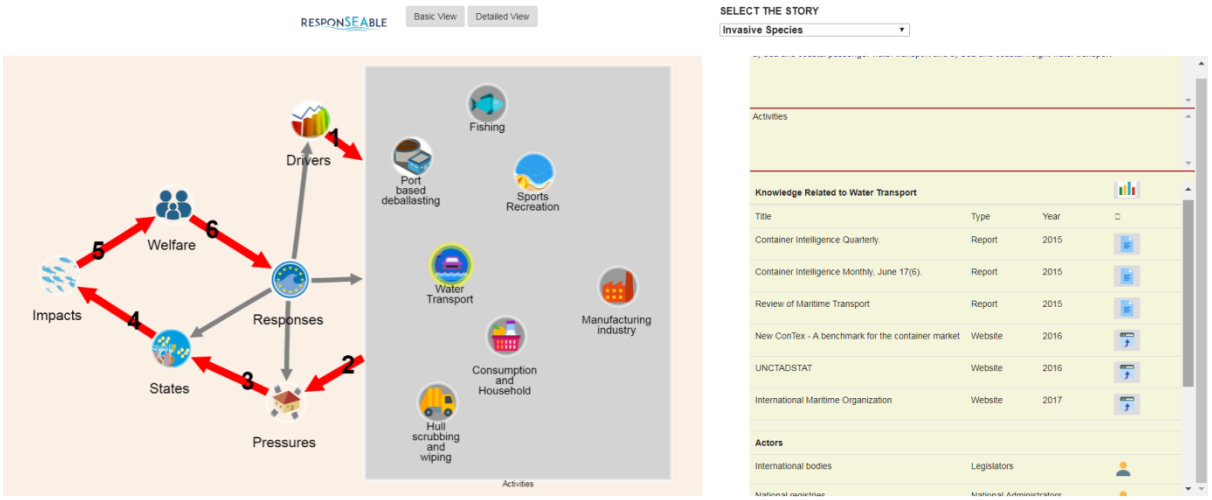



*Figure 6-5: Detail for specific node, e.g. 'Water Transport'*

In the default 'Basic View' the user can expand one or more of the categories to see all of the individual drivers, activities, pressures etc. (see Figure 6-6).

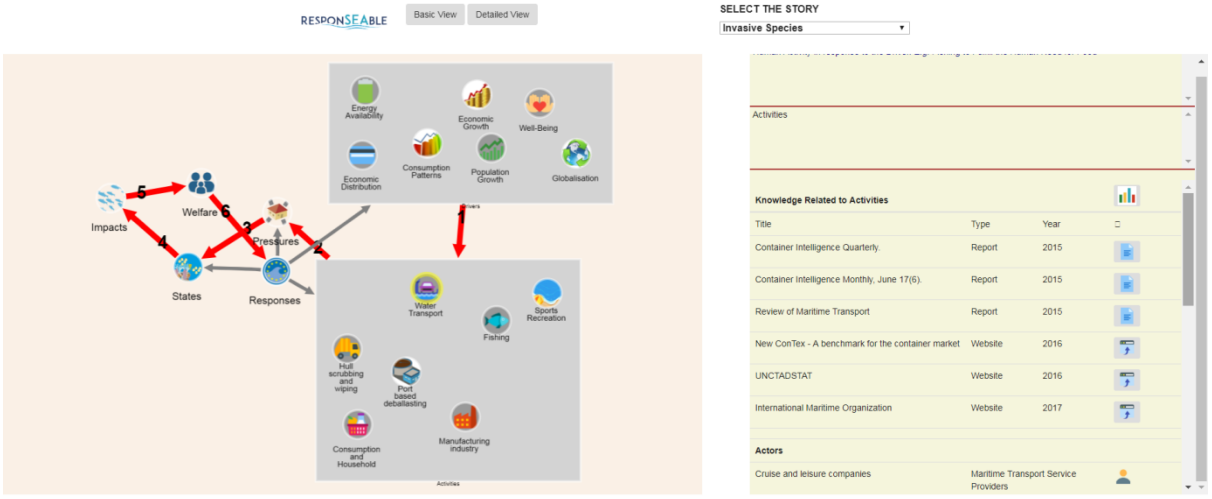



*Figure 6-6: Expanding Drivers, Activities, etc.*

## Links between categories

If the user would like to see all of the knowledge relating to any links between categories, for example knowledge on links between any Activity and any Driver, then they can click on the link between Activities and Drivers. This link will then be highlighted and all the relevant knowledge displayed on the right hand side (Figure 6-7). The same can of course be done for all of the high level links between categories.

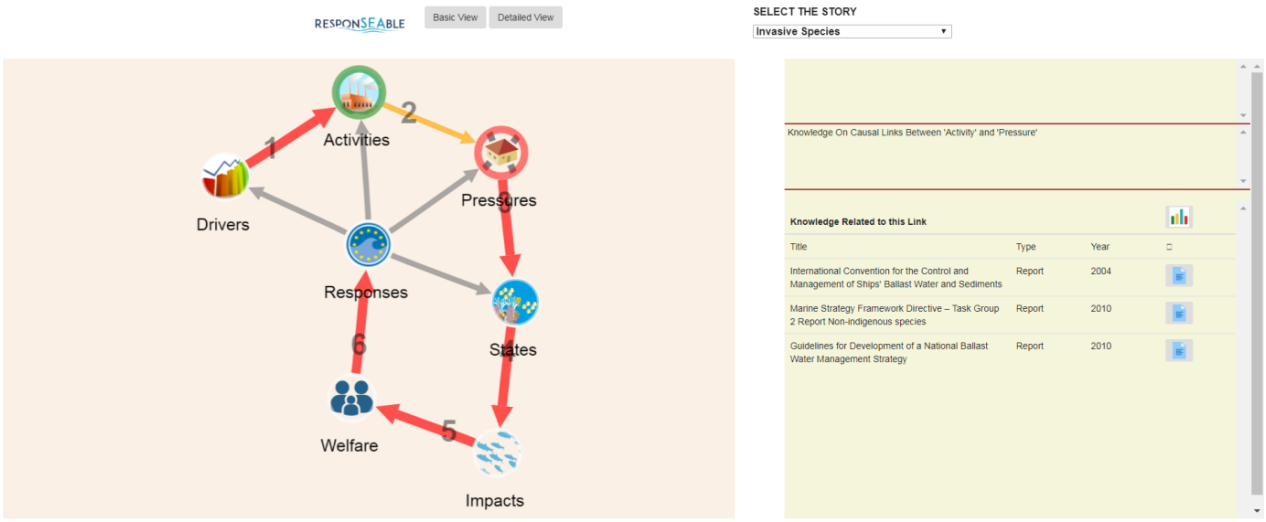



*Figure 6-7: Links between categories*

## Detailed View and Path Knowledge

If the user would like to see all of the links between the individual drivers, activities etc., then they can click on the 'Detailed View' button. All of the links in the DAPSIWR will then be displayed. Clicking on one of the links will cause it to be highlighted, and any knowledge, or evidence, which is attached to that causal link in the DAPSIWR will be listed on the right hand side (Figure 6-8).

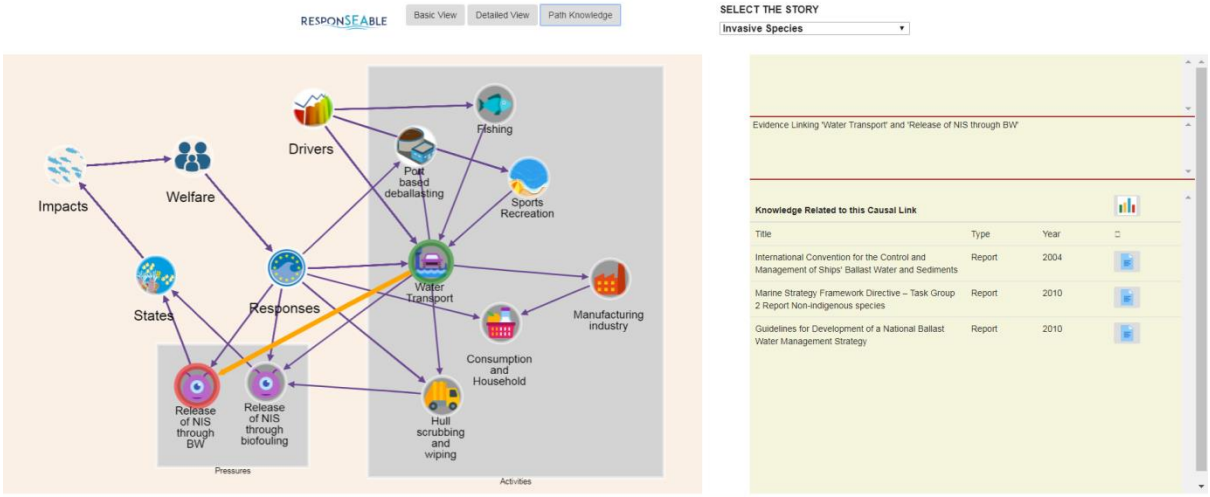



*Figure 6-8: Evidence (knowledge) on links*

## Knowledge On shortest path between any two nodes

When the 'Detailed View' is enabled the user can expand as many of the 'parent' category nodes as they like, potentially revealing all of the links in the DAPSIWR. To find the evidence / knowledge on the shortest path between two nodes, the user simply clicks on the start node (it is highlighted in green) and then the end node (it is highlighted in red), and

clicks on the ‘Path knowledge’. This will invoke the *shortest path* algorithm created by Dijkstra[7], which finds the shortest path between two nodes in a graph.

Once the shortest path is found it is highlighted in yellow, and all of the knowledge which is found attached to the causal links on the path (not the nodes), will be collected and displayed on the right hand side as normal (Figure 6-9).

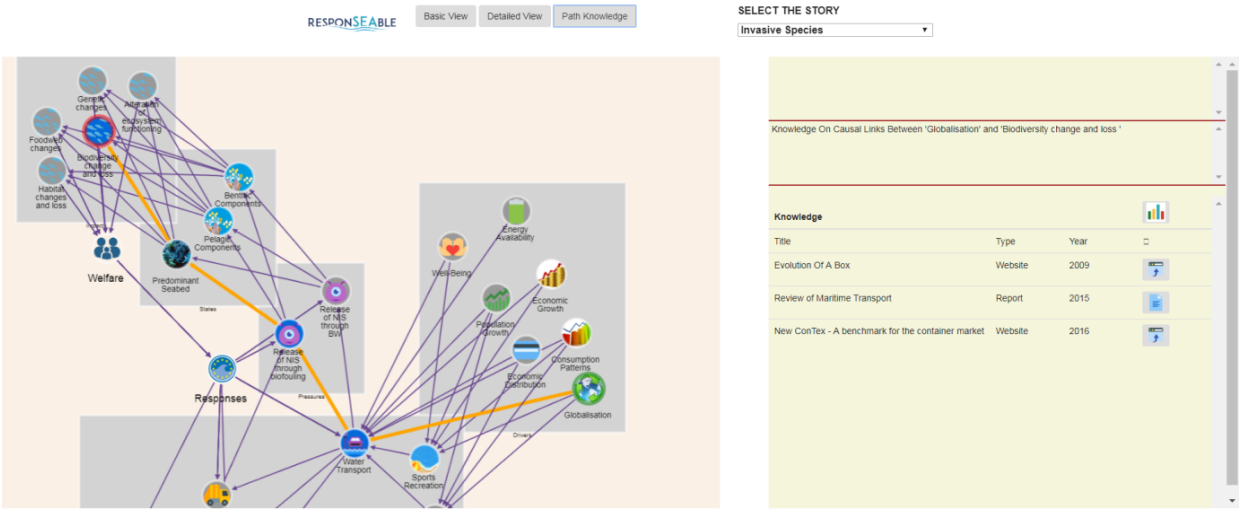



*Figure 6-9: Knowledge on shortest path between nodes*

[7] https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm

# 7. Conclusions

The ResponSEAble Knowledge Base and its applications (Editor, Knowledge Import, Viewer, Reporting and Querying) represent a significant investment on behalf of the project team in realising the vision of an DAPSIWR-based structuring of the knowledge of human-ocean systems, including the economic value chains and actors involved.

The system has enormous potential to be applied not only to ocean topics, but any human-environmental system based on the DAPSIWR framework. In fact, it would be possible to use other frameworks also, with some additional development work. As far as we know, it is the only system of its kind in existence.

The development team are motivated to continue this work and make the platform available on a wider basis. For example, it would be ideal for use in schools- or college-based programs, where students could use the platform to document their ocean / environmental projects using the DAPSIWR framework, gather knowledge on the causal map using the KB, and publish their work using the KB Viewer.

Like never before, public engagement with, and understanding of, environmental issues and human-environmental systems is vital to our welfare and that of the planet.

# ANNEX. Technical Overview

## *Development and implementation of DAPSIWR, Actor and Media Type classifications*

The classification hierarchies for DAPSIWR elements and Media Types are listed in full in the Appendices of ResponSEAble Deliverable 3.1: 'Communication and Perception of the Human-Ocean Relationship'. In the Appendices, in most cases, the source of the classifications is given.

These classifications are listed more formally, together with general properties, in the spreadsheet 2018_02_07_content_classifications.xlsx, developed co-operatively between Tamer Fawzy, Caroline Brennan and George Karavokiros in order to specify the ontologies to be used in the ResponSEAble Knowledge Base. This spreadsheet also includes a classification hierarchy for Actors. Note that the Knowledge ontology in the database uses the Media Types classification.

OrientDB is a graph database that supports SQL-like query syntax and, importantly, definition of a database schema. The classification hierarchies were implemented using an OrientDB schema (defined in the script kb_schema.osql). DAPSIWR nodes are implemented as OrientDB vertices, and DAPSIWR causal relationships as edges between vertices.

Here is a sample of the schema, in which the top-level class 'Framework' is defined (like 'Thing' in an ontology), and Driver subclasses defined:

```
create class Framework extends V
create class Driver extends Framework;
create class IndirectDriver extends Driver;
create class DirectDriver extends Driver;
create class DemographicDriver extends IndirectDriver;
…
```

The whole DAPSIWR framework is built by extending the above Framework class. The Story class and the Actor and Knowledge hierarchies are built in a similar way, with properties as appropriate. Most classes have simple properties: name and description. The Knowledge class has the most properties, including some embedded properties (embedded Actors for target audience, and an embedded Actor for publisher). Here are a few more sample lines:

```
create class Story extends V
// properties
create property Story.name string
create property Story.description string
create property Story.topic string
..
create class Actor extends V
// subclasses
create class GeneralPublic extends Actor
create class Professional extends Actor
create class PrimarySector extends Professional
..
create class Knowledge extends V
// properties
```

```
create property Knowledge.knowledgeId string
create property Knowledge.keyWords embeddedlist string
create property Knowledge.targetAudience embeddedlist Actor
create property Knowledge.publisher embedded Actor
..
// subclasses
create class FirstDegreeMedia extends Knowledge
create class VisualArts extends FirstDegreeMedia
..
```

The script kb_schema.osql creates the database 'kb' and its schema. A second script, kb_data.osql, connects to the database 'kb', creates instances of Users, and populates the 'kb' database with 3 test stories. Here is short extract from kb_data.osql, in which the Story vertex and two Driver vertices are created, and an edge between the Story and the first Driver are created:

```
create vertex Story
set name='TEST Microplastics and cosmetics in the EU',
topic='Microplastics',
region='EU',
description='..',
..
// Drivers
create vertex ConsumptionPatterns
set name='Consumption Pattern',
description = 'More and more personal care and cosmetic products
contain microplastics..'

create vertex MaterialAvailability
set name='Material Availability',
description = '..'
..
// Link to Story
create edge
from (select from Story where name='TEST Microplastics and
cosmetics in the EU')
to (select from ConsumptionPatterns) set description='Story starts'
```

User classes are also defined in the Orient DB scheme and implemented as OrientDB vertices. Users can have edges to Stories, if they created the stories or can read them. User logins are handled using User vertex properties.

## *How clients create and edit Stories in the database*

The Java project contains a stateless Resource class which defines a large number of RESTful web services to support client interaction with the OrientDB database on the server. The web services call methods in a Singleton EJB which uses the Java Graph (Tinkerpop) libraries to query the OrientDB database and create responses in JSON to send back to the client in the web service response.

When a Story is created in the Editor, the client calls web service methods to create vertices and edges on the server and specify their properties. Similarly, when editing Stories, clients send web service calls to add, change or delete vertices and edges in the database.

## *How Story data is passed to clients from the database*

More web services exist to support a wide range of queries on the Story data. Here are the first few lines of an example:

```
// Get representation of a whole (single) Story from the graph //
database (VERTICES and EDGES), in JSON format

@GET

@Path("/storyk/{id}")

@Produces(MediaType.APPLICATION_JSON)

public String getStoryK(@Context HttpHeaders headers,

@PathParam("id") String storyId) {

      …
```

These web services return JSON to the client. Here is a short extract of the JSON for the IAS Key story:

```
{

  "mode": "NORMAL",

  "vertices": [

    {

      "name": "IAS Key Story",

      "topic": "The story of Invasive Alien Species in the EU",

      "description": "The sea and the coasts have always been
drivers of the economy...",

      "synopsis": "Introducing Invasive Alien Species through
Ballast Water and Hull Fouling",

      "region": "Europe - Mediterranean Sea",

      "definitions": "",

      "_id": "#14:8",

      "_type": "vertex",

      "_label": "Story"

    },

    {
```

```
  "name": "Globalisation",
  "description": "Globalisation",
  "comment": "Globalisation",
  "label": "Globalisation",
  "_id": "#54:0",
  "_type": "vertex",
  "_label": "Globalisation",
  "category": "Driver"
},…
```