MDEnet | The home of model-driven engineering

Facilitating provenance documentation with a model-driven-engineering approach

MDEnet Seedcorn funded project

Aston University and Earthwatch collaboration

Aston University BIRMINGHAM UK

earthwatch EUROPE

How do you evaluate the quality of a dataset?

- Before using a dataset you should carefully judge its suitability
  - BIAS?
  - Poor handling during processing?
  - Bad sampling techniques?
- To assess these things a dataset needs accompanying documentation
  - Metadata
  - Provenance, lineage
- **FAIR** - **F**indability, **A**ccessibility, **I**nteroperability and **R**euse
  - *"Good"* documentation/metadata makes a dataset more reusable

What would you expect to see in the documentation of a dataset?

- What is *"good"* documentation or metadata?
- Includes everything (obviously but that is not practical)
  - Processes, algorithms?
  - Sources, citations?
- Standards help outline what should be included
  - ISO 19115-3 -> ISO 19157 specifically quality, provenance/lineage
  - W3C PROV
  - *And there are likely many more we would like to know about*

Standards can be hard to use by non-experts

- ISO19115-3 is detailed, therefore, complex and hard to understand
- W3C-PROV is much simpler, but doesn't provide detailed structures
- Technical implementations are a challenge
  - Standard are commonly associated with data format structures
    - e.g. ISO19115-3 defines an XML file format
  - Data format structures are not always human friendly
    - Usually accompanied by a (large) document explaining them
- Citizen Science
  - Struggles with how to document datasets produced by a project
  - Skills and expertise to use a standard might not be available

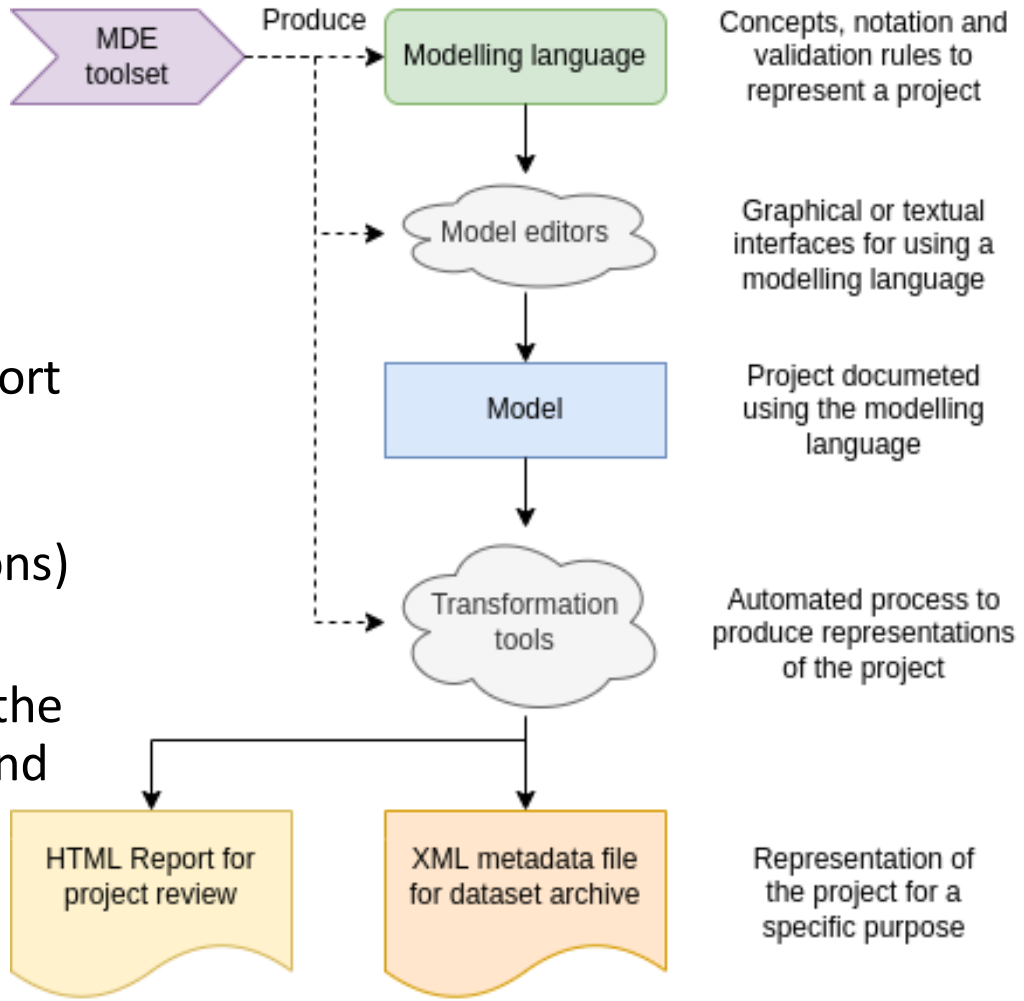
Assistive Software solutions

- Documentation could be like a dynamic application form
  - Questions and steps to follow to document the dataset creation
- Software tools based on standards could help
  - Create guided processes to documenting a project
  - User interface could provide feedback on errors/missing information
  - Output metadata files in multiple formats (XML, JSON* etc.)
- Writing *good* software is **hard** with traditional programming techniques
  - User interfaces need creating
  - Logic for User input validation
  - Formatting data for output to files
  - Maintenance challenges, updating/extend with changes to standards

**We are aware of some proposed/emerging JSON implementation of existing standard*

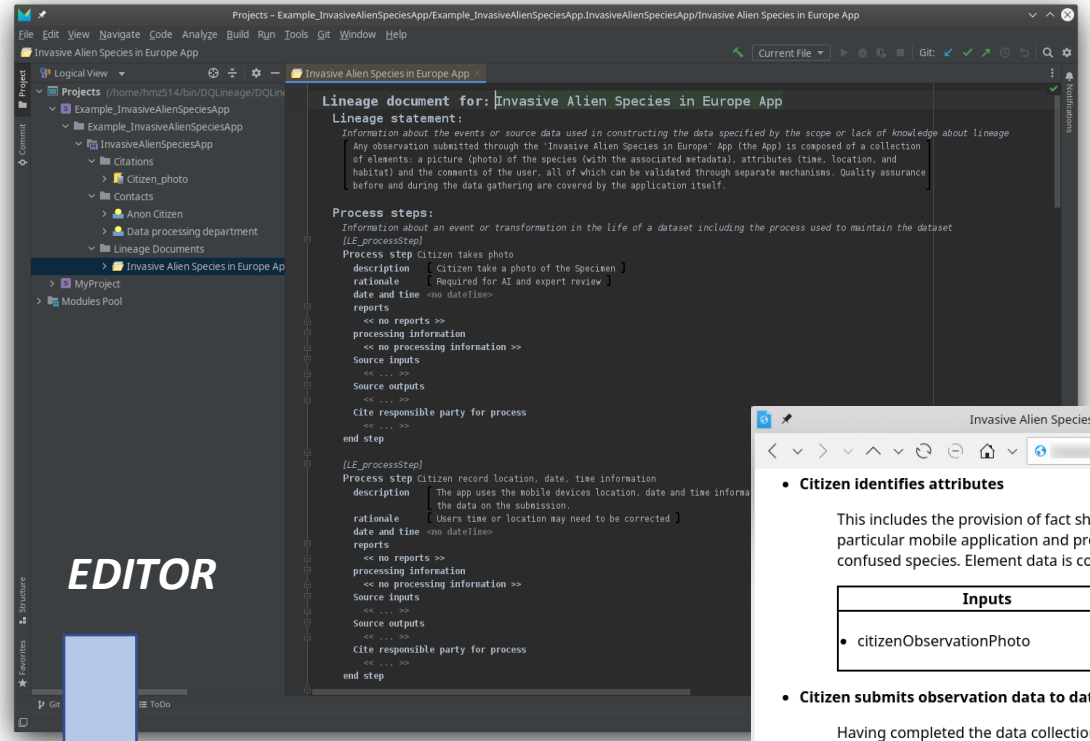
Model-driven engineering (MDE) approach

- Uses Modelling languages to support engineering tasks
  - Such as creating model representations of system *(there can be many depending on perspective)*
- Modelling language: Concepts, notations and validation rules for creating a model
- MDE tools exist to create modelling languages and support operating on models with the languages created
  - Editing, validating, transformation/generation of models, code and digital artifacts (file representations)
- E.g. An existing standard informs the creation of a modelling language using an MDE toolset. This enables the creation of model editors with user-friendly notations and automated transformation into other representations

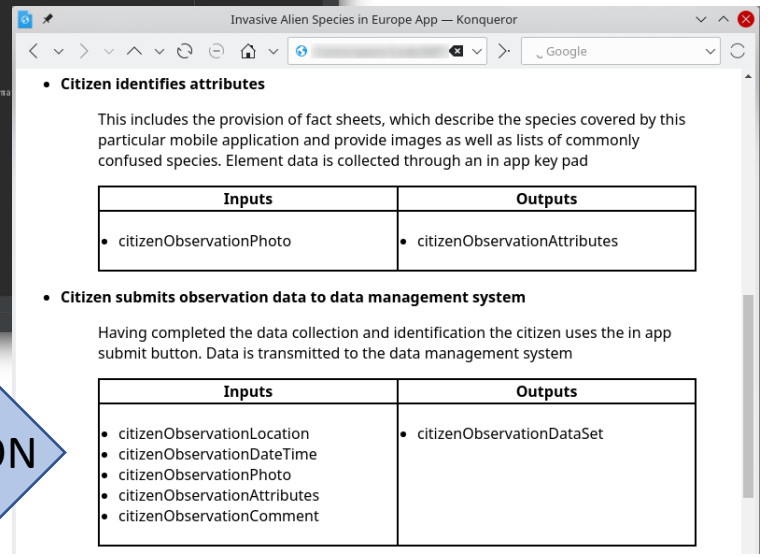


Prototype using Jetbrains MPS

- Modelling language based on ISO 19115-3
- "Text like" editor for documenting (modelling) a project
- Editor features:
  - User input validation
  - 'Advice' and descriptions
  - Automation for some common tasks
- A (model) representation can be used to generate multiple outputs:
  - Report documents (HTML)
  - Metadata files (XML)



AUTOMATED TRANSFORMATION



REPORT STYLE OUTPUT

https://www.jetbrains.com/mps/

Early user reception

- Features inherited from Jetbrains-MPS affect
  - Environment for software developers (Not a friendly user environment)
  - Historically it is used to create modelling languages for generating Java code
  - \- Not web-based so has software setup overheads and dependencies (Java/Git)
  + Enable the creation of a prototype quickly with advanced editor features and transformations (user input validation, user advice, generate HTML)
- User Experience of prototype tool: Notation/principles/features
  + Users can "sketch out" documentation, mandatory fields don't block this process
  + Users would like additional editor views such as workflow diagrams *(this is possible)*
  - ***Terminology*** in the specification in some instances doesn't fit the context of editor activities e.g. data item might be a better description than data source *(can be changed)*
    - Outdated terminology inherited from the standard
  + Highlights errors in "red" to prompt to correct issues (validation) (- doesn't explain what the problem is)
  + Once keyboard commands and navigation are known, they become intuitive
  + Prototype editor could use more descriptive hints or pop-ups on sections (More help with terminology, clearer on mandatory/optional information)
  + Reusable concepts e.g. contact details, citations, steps, processes, data items etc.

Conclusion and future work

- User acceptance testing with Citizen Science project managers (to do)
  - Collect feedback on the approach to guide future developments
- Modelling language refinement
  - Investigate other standards or merge concepts from several standards
  - Controlled vocabularies to inform 'drop-down' or 'select from' lists
- Investigate other forms of output
  - Transformations to different standards and representations
  - Human readable forms (Workflow diagrams etc.)
- Web-based implementation
  - Collaborative workspace
  - No software setup required
  - Multiple editor views of the project model (graphical and textual)
  - User interface development
- Tools and training are needed for the unavoidable complexities
  - Friendly notations, editors and user training to improve dataset documentation

Thank you for attending

Please contact us if you are interested in these ideas!

- Owen Reynolds
  180200041@aston.ac.uk
- Lucy Bastin
  l.bastin@aston.ac.uk
- Antonio Garcia-Dominguez
  a.garcia-dominguez@york.ac.uk
- James Sprinks
  jsprinks@earthwatch.org.uk

Aston University
BIRMINGHAM UK

https://aston.ac.uk/

MDENet | The home of model-driven engineering

https://mde-network.com/

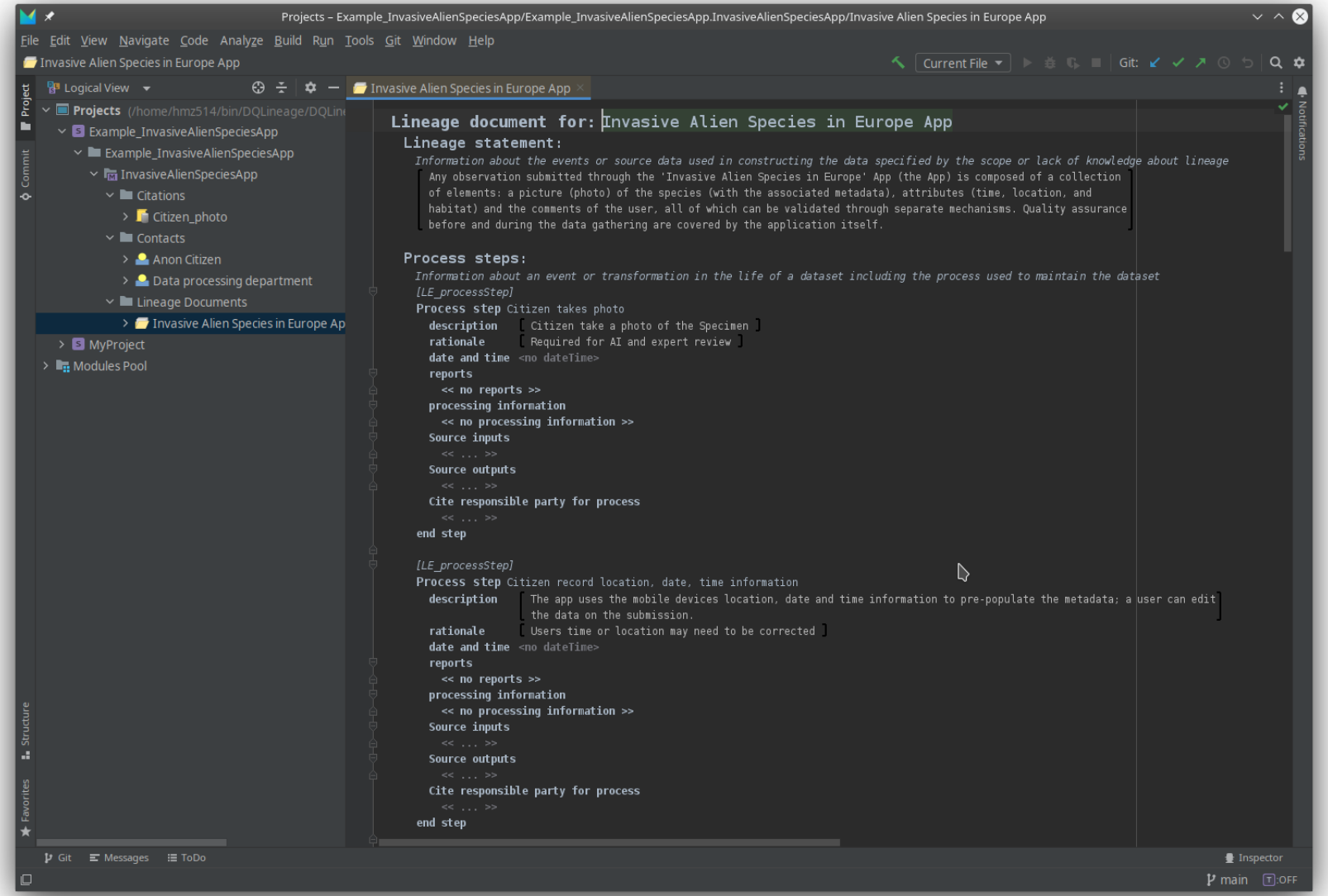
earthwatch
EUROPE

https://earthwatch.org/

Appendix

Additional slides with some more detailed information and examples

MPS Editor

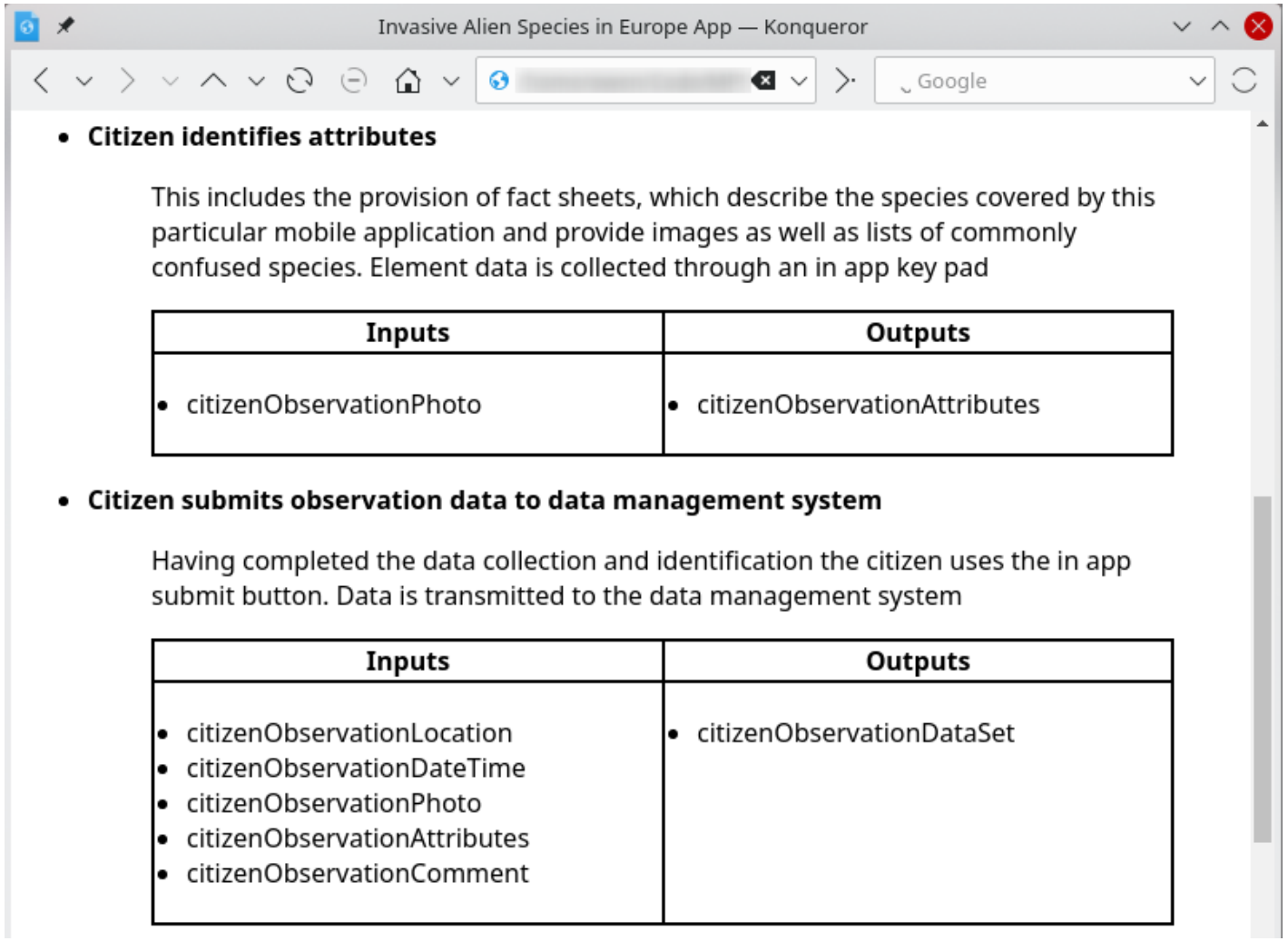


The editor show is generated by MPS and can be customised to some extent.

To develop the application, the concepts for the modelling languages were defined based on ISO19115-3. Then editors were defined for each concept. Editors for concepts are then nested to create the Lineage document structure seen in the editor.

E.g. The concept of an address is defined with an editor. Where ever an address is needed the concept/editor can be connected

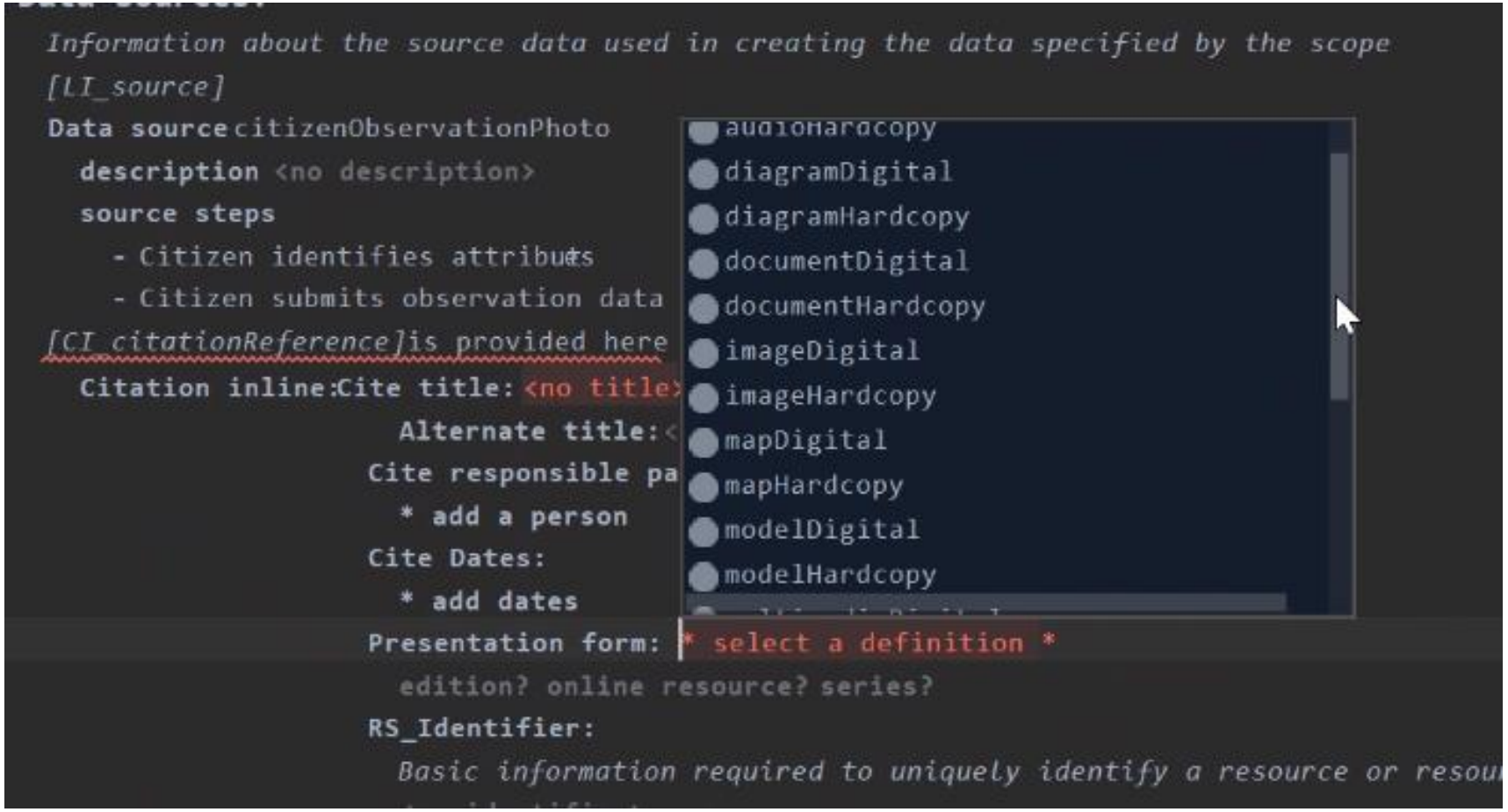
Example HTML output



This example report is rendered in HTML, it is possible to create other text based notations e.g. Latex.

The modelled project is transformed to produce a representation of the project, which can omit unrequired information from the project. (This example report does not contain everything in the Example project)

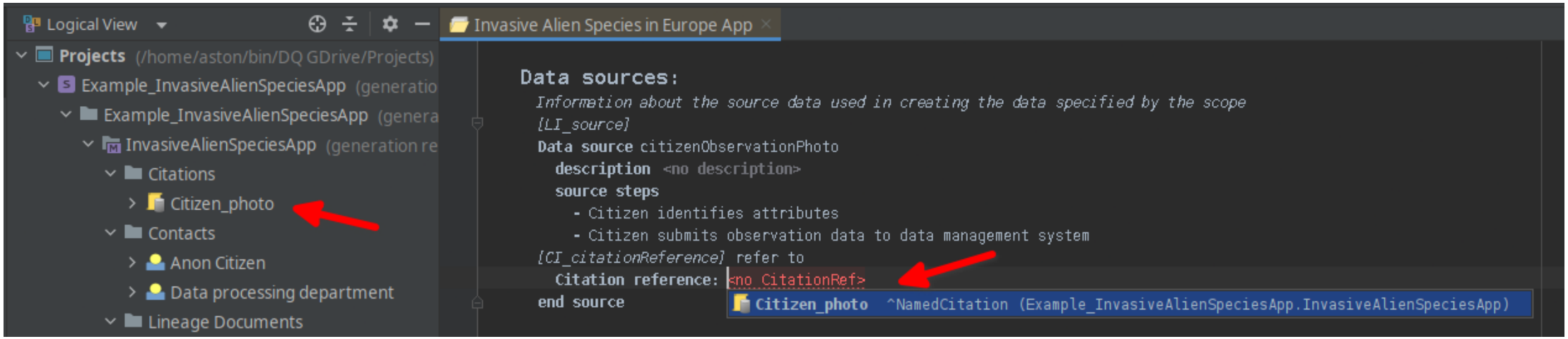
Tool tips and advice



Pressing CTRL+Space on a cell reveals a list of possible inputs. In this example the ISO19115-3 codes for a Presentation form are given. The user can then select as appropriate. Alternatively, the a user can start typing in a cell and completion options are presented.

Cell with red text indicate a problem. However, the nature of the error isn't always clear. This is something that could be improved

Model once and reuse



Some information in a project needs to be referred to more than once. While developing the prototype we identified contact information and citations as two concepts that would reoccur in the documentation. As such, a Named Citation/Contact can be created and referenced from within the lineage document. Thus, the information for a citation/contact can be managed in a single place and consistently represented through the document.

Future development would likely provide similar ways to handle processes and data items.