PyBIRD AI Overview

PyBIRD AI allows users to edit the core content of BIRD in one place, and store the edits locally, or share them under version control for peer review.

Currently much of the information in BIRD is available in a read only format, and it is not possible for users to make, test and share local edits in an organized way.

PyBIRD AI allows users to create a single sqllite database file representing the information exposed by the BIRD website (which is read only) and from the version of the BIRD Input Model and LDM in SQLDeveloper (which is easy to local edits)

PyBIRD AI also has some key processes to derive information related to transformation rule from that information (or from locally edited information) and store it back into the databases

In storing the information, we use an intermediate format used by millions of developers called Python Django. This artefact is a very good source for further automated proceses. This means that with a few commands we can create a usable website for simple updates, a Web API that can be extended, a database that can be queried, an ORM (a means to deal easily with databases in Python code). AI code assistants are also good at helping create Django code because there are so many examples of it in its training.

This moves us towards a goal where we can make a change locally to BIRD and then automatically run executable transformations in Python against a test suite so that all changes can be tested against a test suite to check for regression errors introduced by the change.

Regression errors are very costly in regulatory reporting solutions, and should be found immediately upon changing transfomration, by the person who made and understood the change, not left for secondary users to find, perhaps in production. Changes should only be accepted once it has been checked that they do not introduce regression errors.

Executable transformations are out of scope of BIRD although the published information around transformations (for Finrep) is very detailed. PyBIRD AI can enable an open, well tested, reference implementation of executable transformations which can give the level of confidence in the non-executables transformations , this confidence is required by banks to make use of them.

Note that we can consider the idea of input concepts (e.g input layer and LDM) and output concepts (e.g. datapoints).. we also note that different data dictionaries might be used for input concepts and output concepts (e.g. SDD and EBA)

We need analysis to convert between input concepts and output concepts.

We also may need analysis to translate between dictionaries when input and final output use different dictionaries

BIRDs current primary form of storing this analysis is called’ ‘Mappings’ which are created with detailed manual analysis. We note that many of the transformation concepts (output layer cubes, ‘combinations’ of cell level filters ) are derived automatically from these mappings since the hard work of analysis has been done manually. This may change in future.

We note that in other regulatory systems this primary analysis is done in different ways, and stored in different places according to the methodology , and may have different output layer concepts.

In agile BIRD we follow the approach of using mapping analysis, but this can change to allow for different methodologies that deal with input layer/output layer concepts and dictionaries.

The methodology for such analysis should be clearly defined in BIRD, and when it has a technical form this should be well exposed for users to edit it and run processes to find the consequences (intended and unintended) of editing this analysis (such as the new output layers or combinations that would be generated from an edited mapping)

Currently the mappings, and the knock-on effects of changing mappings are read only content in BIRD and users have no ability to determine the effects on transformations if these are changed. AgileBIRD changes this to open up the process and will continue to do so if the analysis methodology changes after review.