Supporting Undo and Redo in Scientific Data Analysis

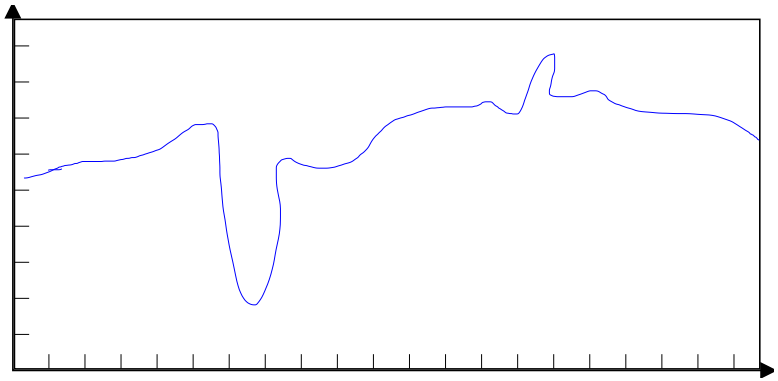
Xiang Zhao[UM], Emery R. Boose[Harvard], Yuriy Brun[UM],
xiang@cs.umass.edu, boose@fas.harvard.edu, brun@cs.umass.edu

Barbara Staudt Lerner[MHC], Leon J. Osterweil[UM]
blerner@mtholyoke.edu, ljo@cs.umass.edu

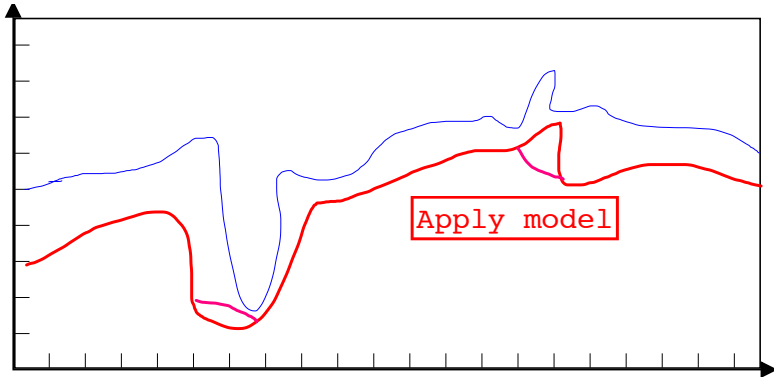
[UM] University of Massachusetts Amherst
[MHC] Mount Holyoke College
[Harvard] Harvard University

http://laser.cs.umass.edu

Scientific Data Analysis



Raw Data

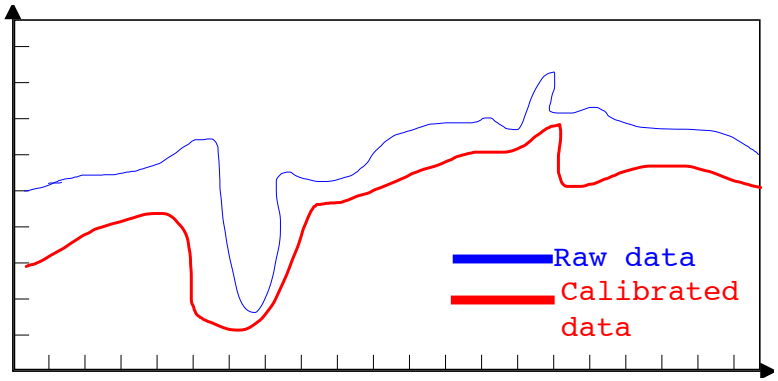


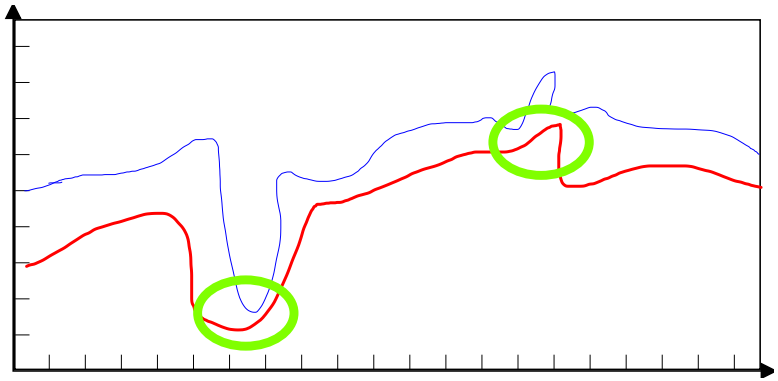
Gap-filled Data

↓

↑

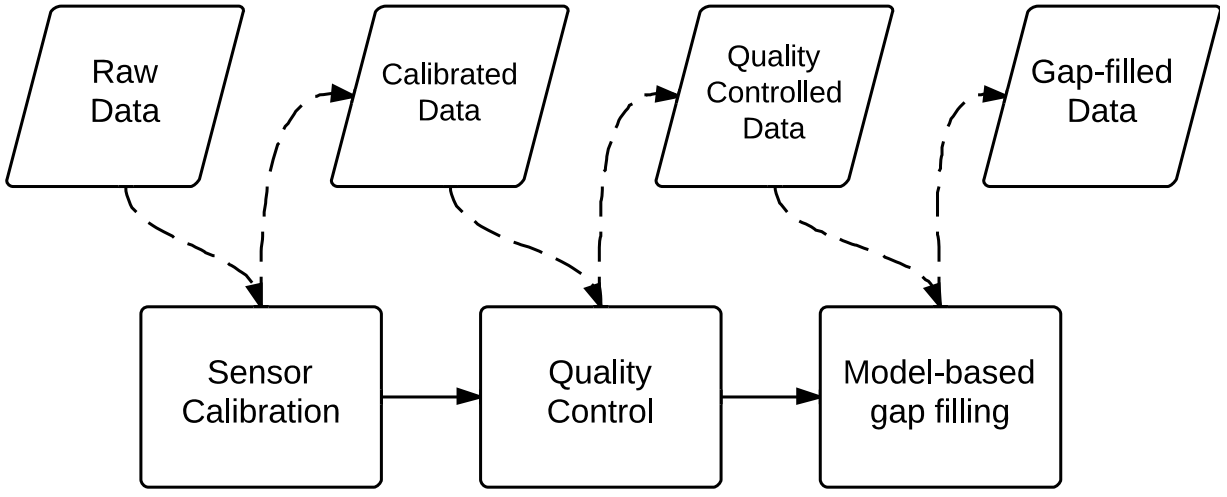
Calibrated Data → Quality Controlled Data



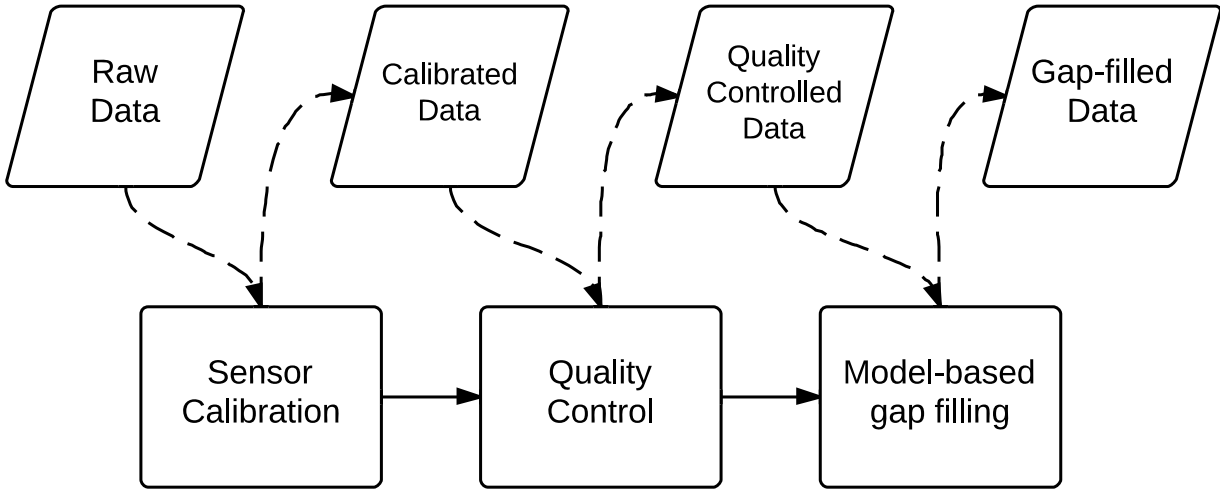


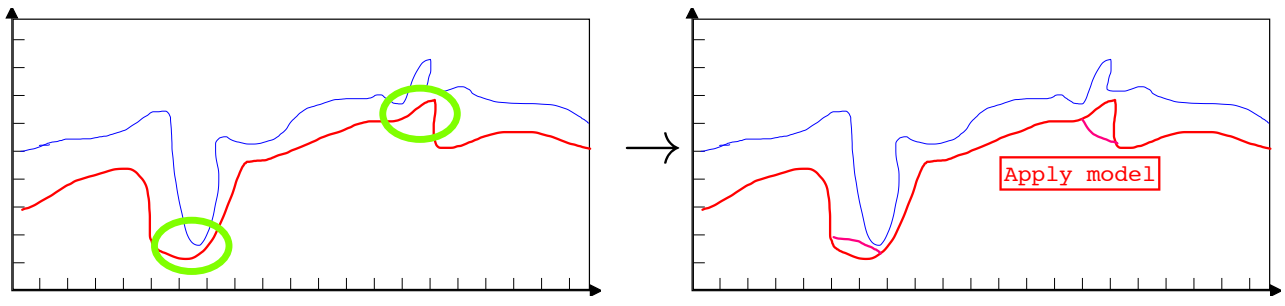
Scientific data goes through a series of complex transformations.

Undo and Redo in Scientific Data Analysis



Undo and Redo in Scientific Data Analysis

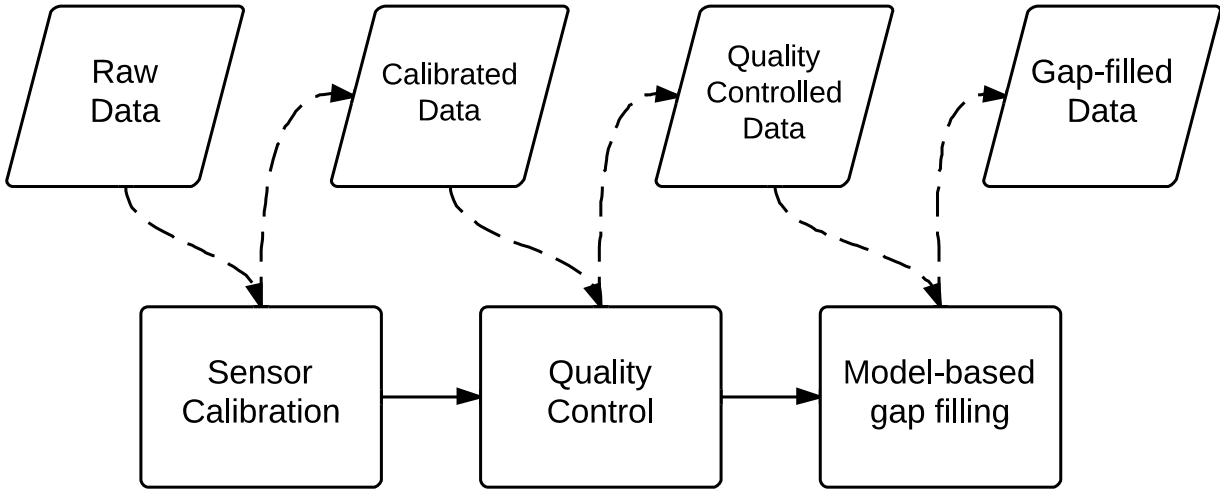


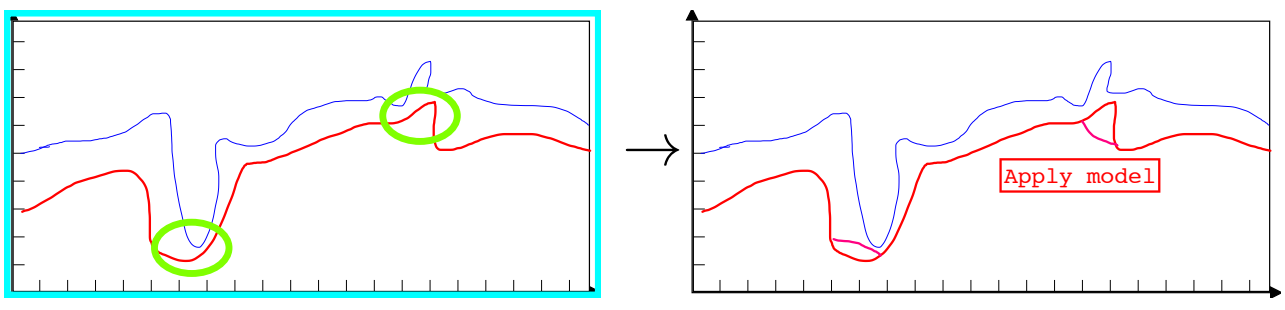


→apply another model?

- Transformations may be revisited as more information is available.

Undo and Redo in Scientific Data Analysis

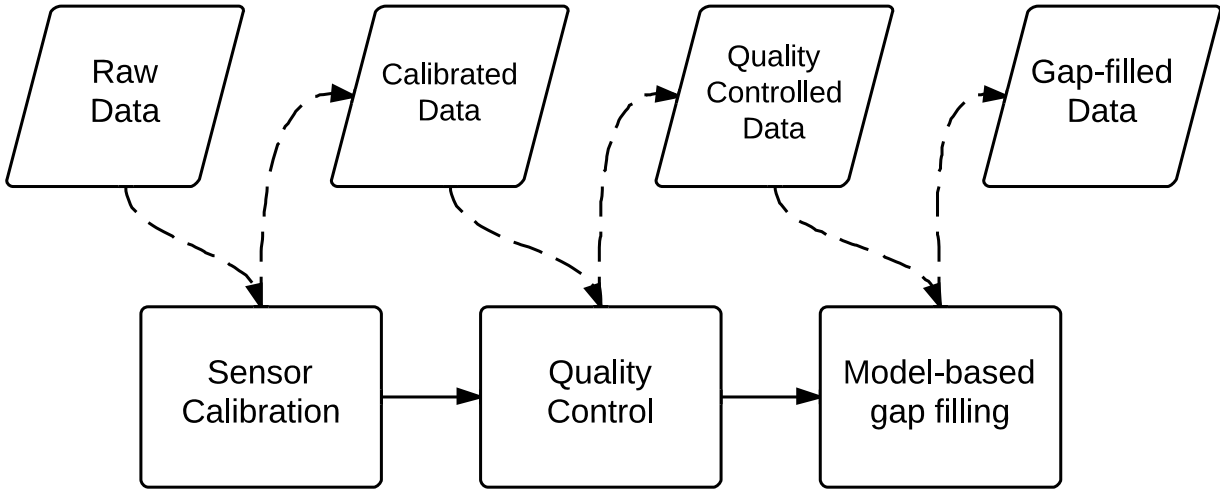


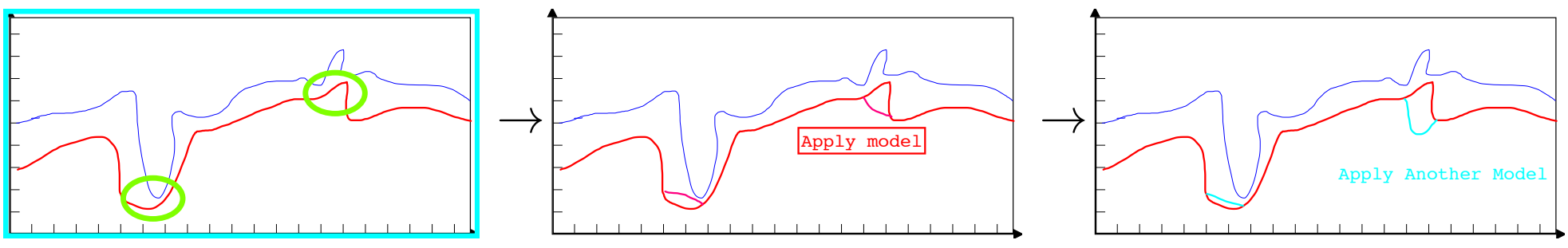


→apply another model?

- Transformations may be revisited as more information is available.

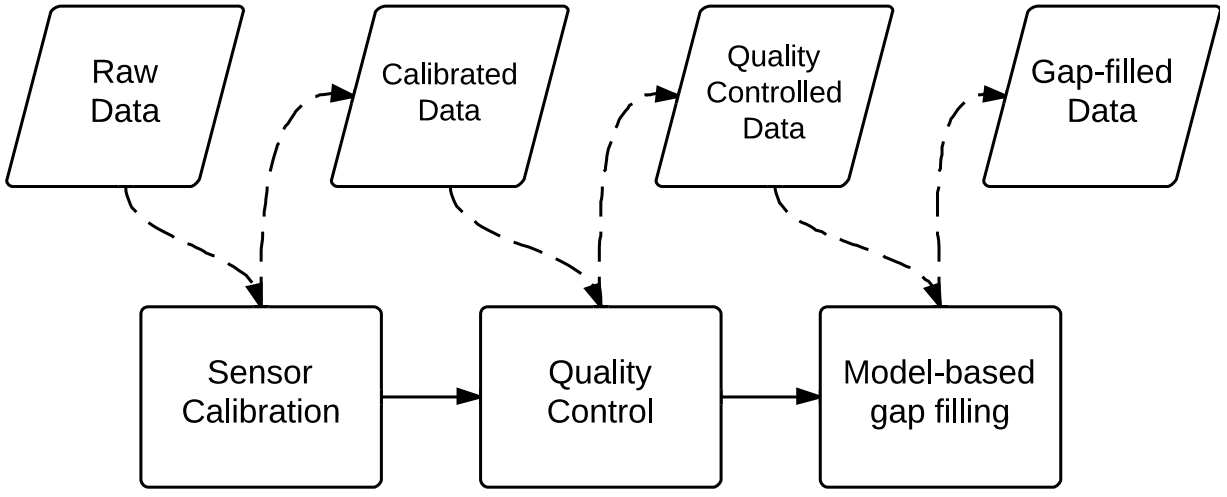
Undo and Redo in Scientific Data Analysis

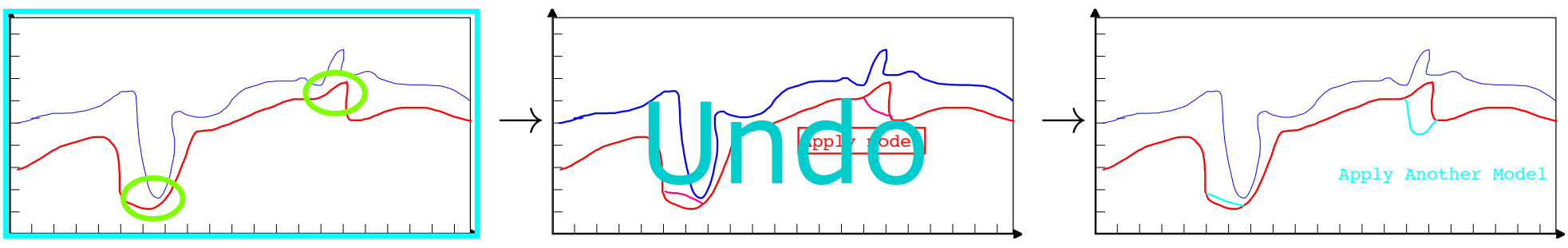




- Transformations may be revisited as more information is available.

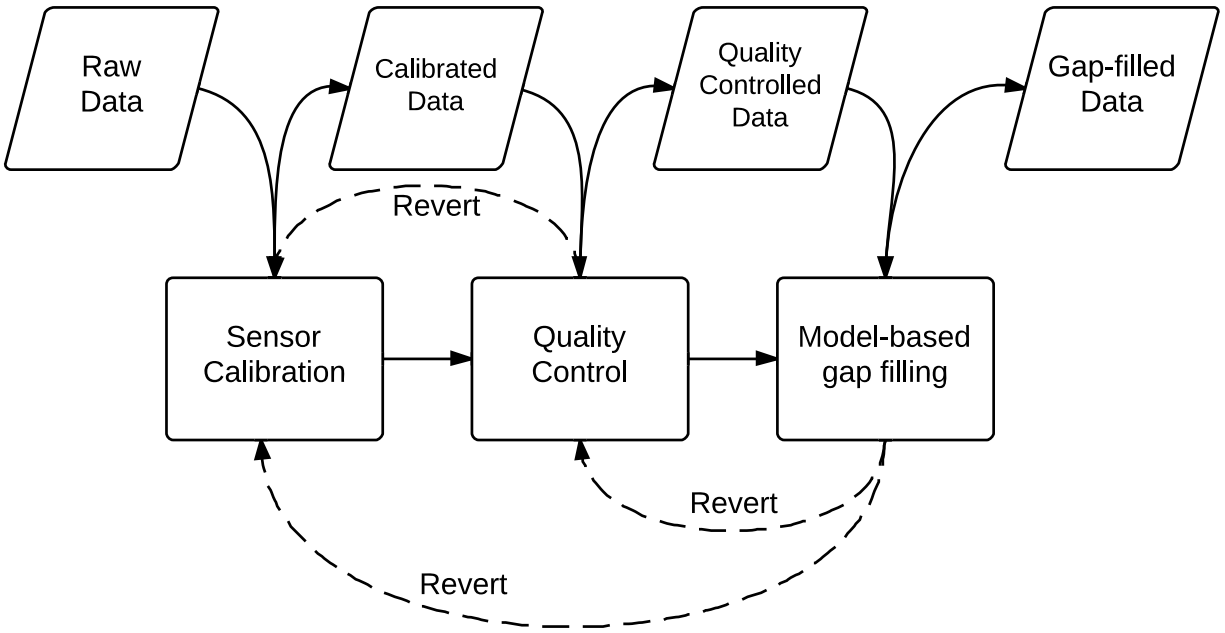
Undo and Redo in Scientific Data Analysis

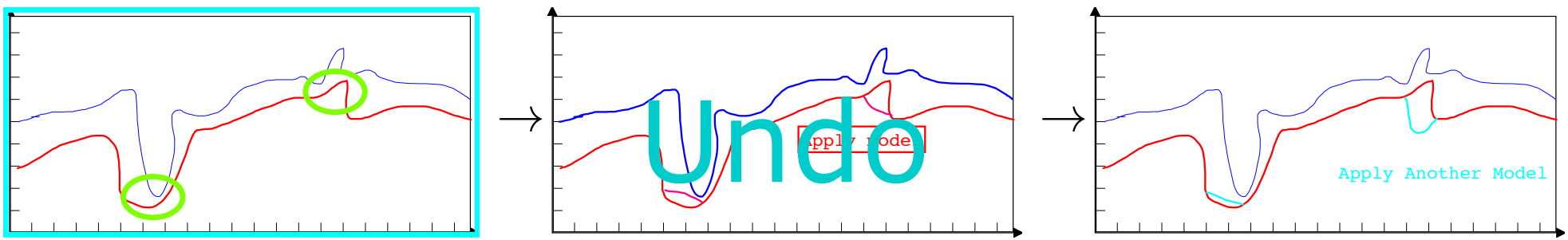




- Transformations may be revisited as more information is available.

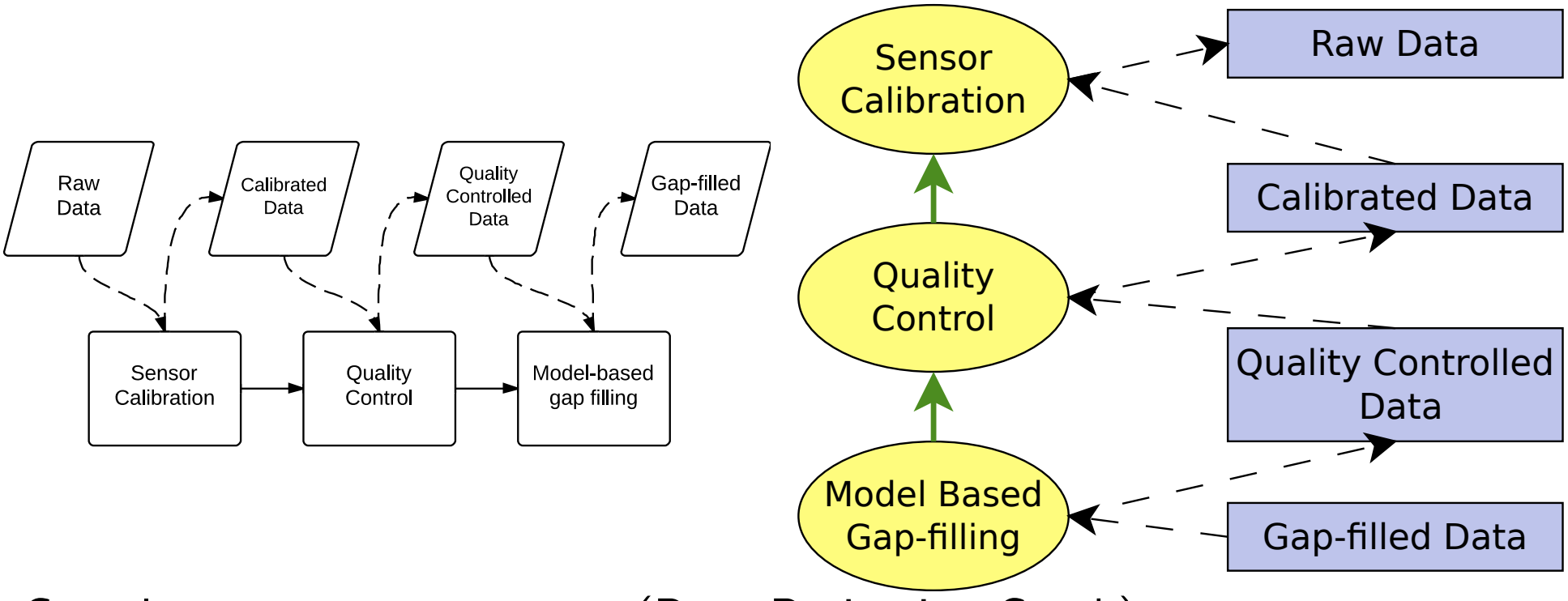
Undo and Redo in Scientific Data Analysis





- Transformations may be revisited as more information is available.
- Undo and redo happen often
  - Undo and redo should not cause restarting from scratch.
  - Intermediate computations need to be taken advantage of.

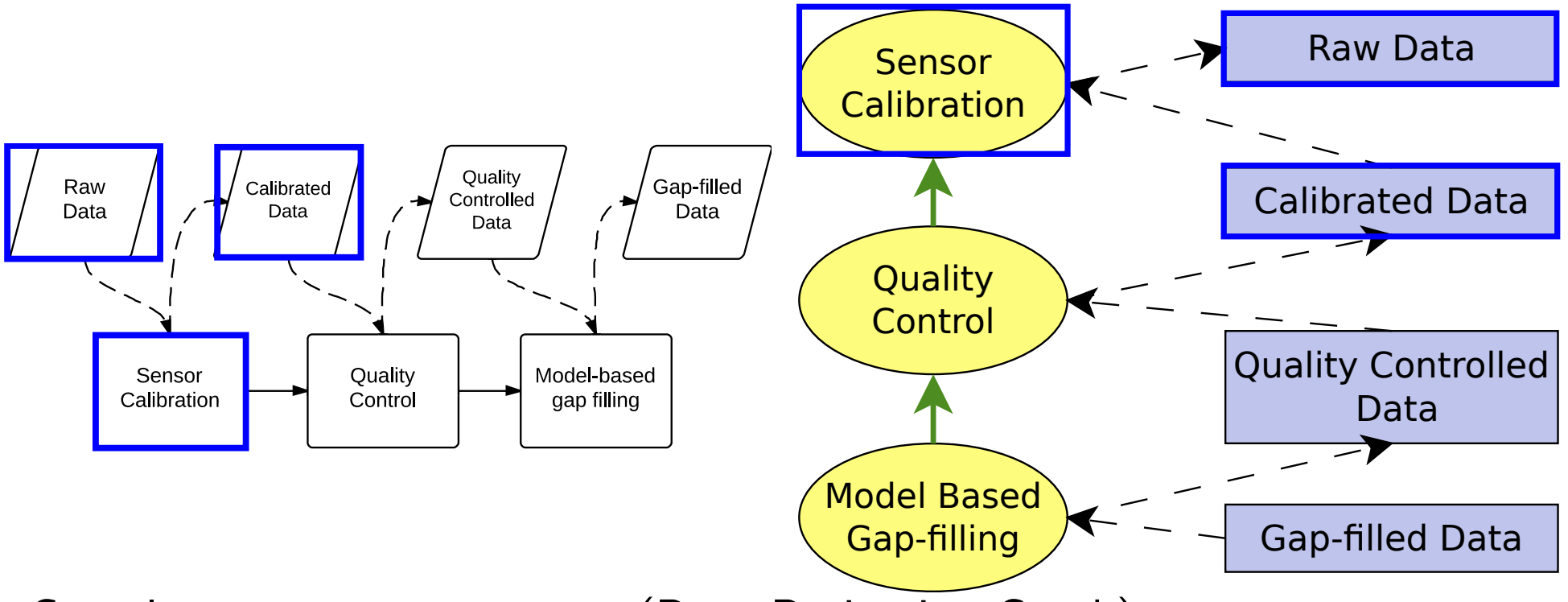
DDG: Provenance Support for Undo/Redo



Complete process provenance (Data Derivation Graph)

- Automatically records detailed process execution history
  - data creations and modifications
  - step execution sequences

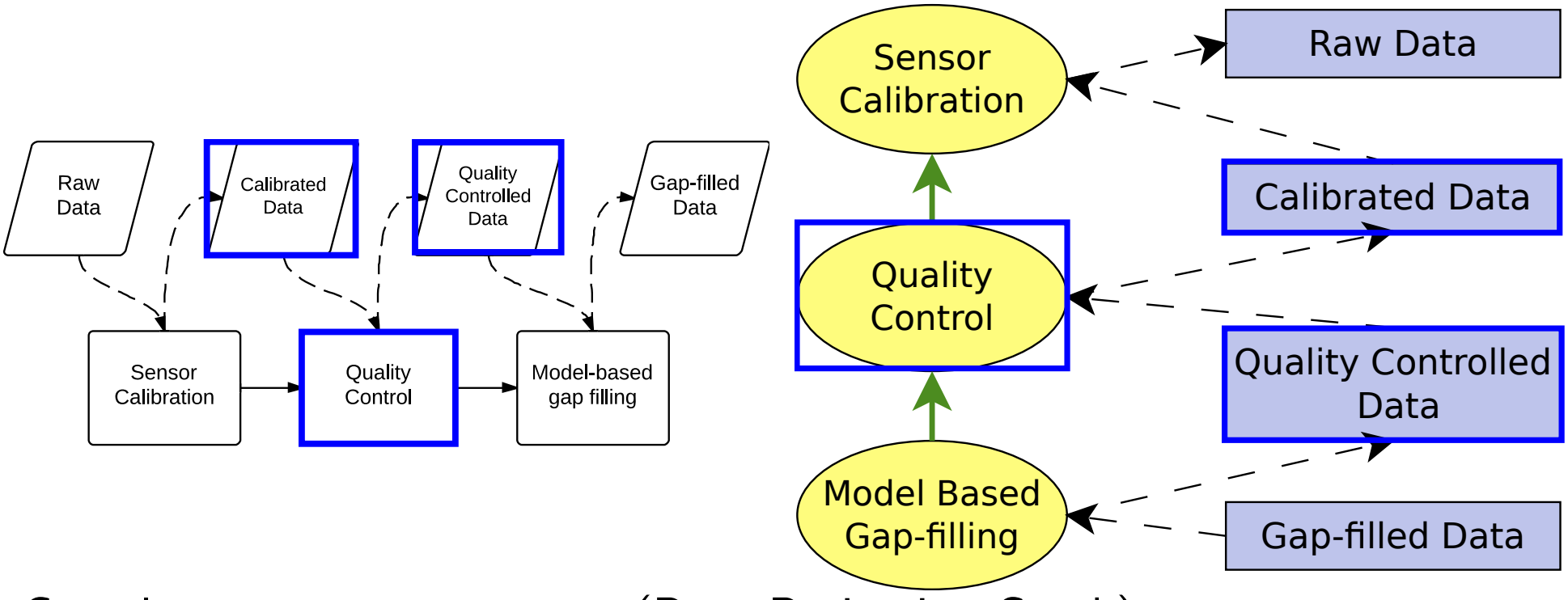
DDG: Provenance Support for Undo/Redo



Complete process provenance (Data Derivation Graph)

- Automatically records detailed process execution history
  - data creations and modifications
  - step execution sequences

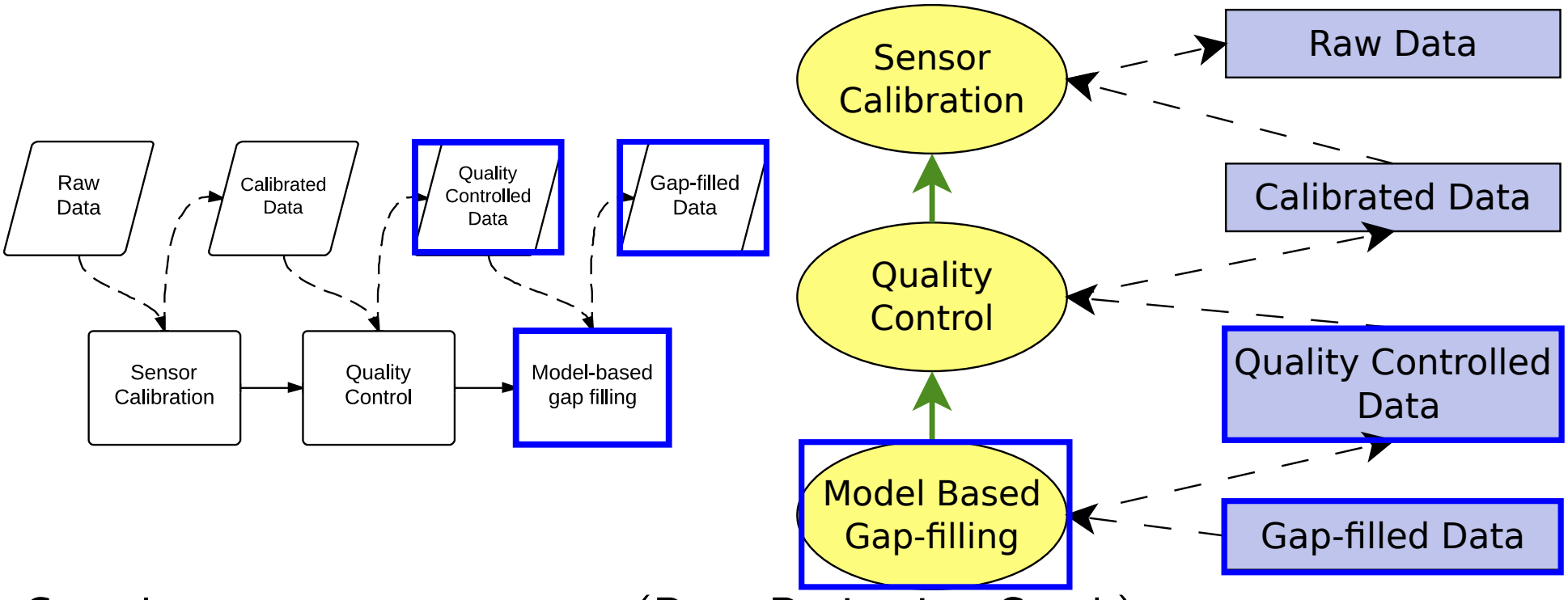
DDG: Provenance Support for Undo/Redo



Complete process provenance (Data Derivation Graph)

- Automatically records detailed process execution history
  - data creations and modifications
  - step execution sequences

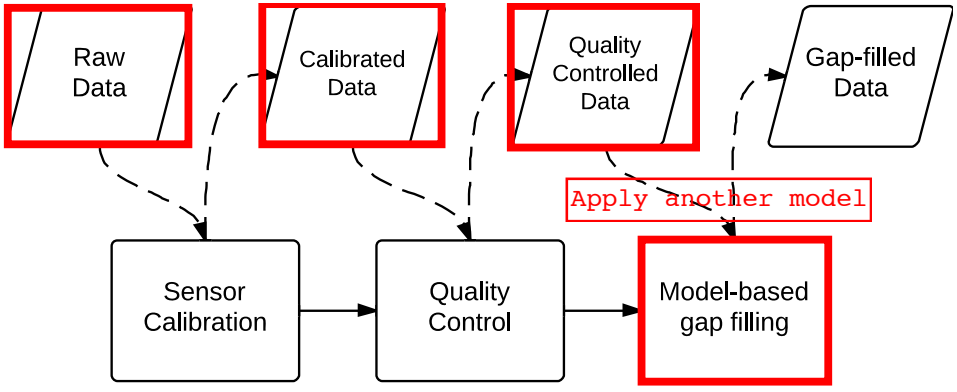
DDG: Provenance Support for Undo/Redo

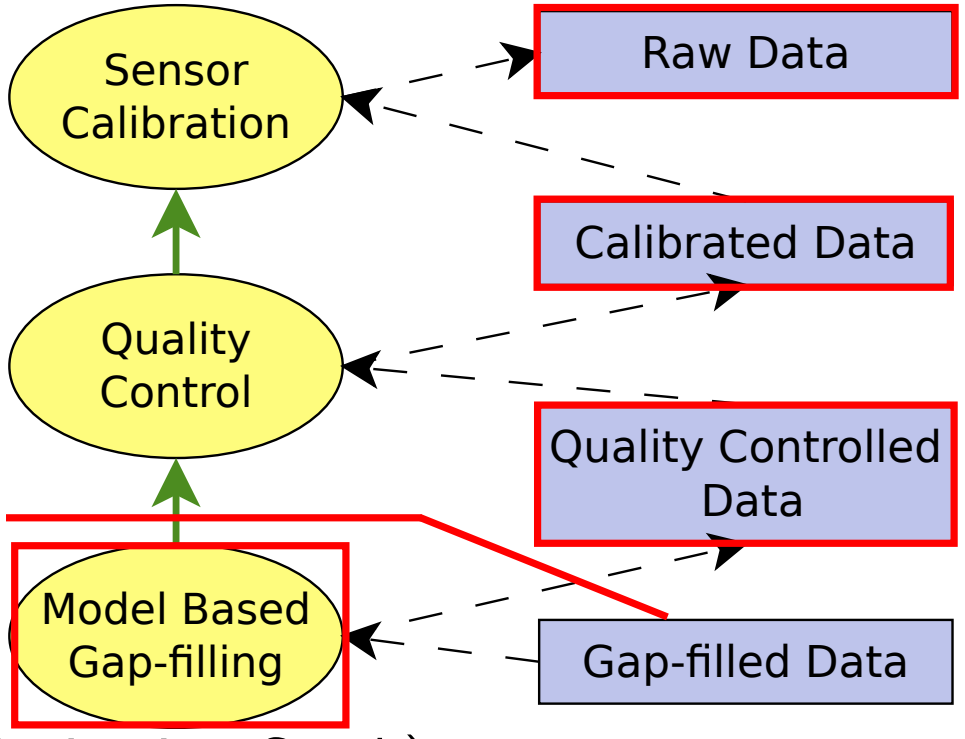


Complete process provenance (Data Derivation Graph)

- Automatically records detailed process execution history
  - data creations and modifications
  - step execution sequences

DDG: Provenance Support for Undo/Redo

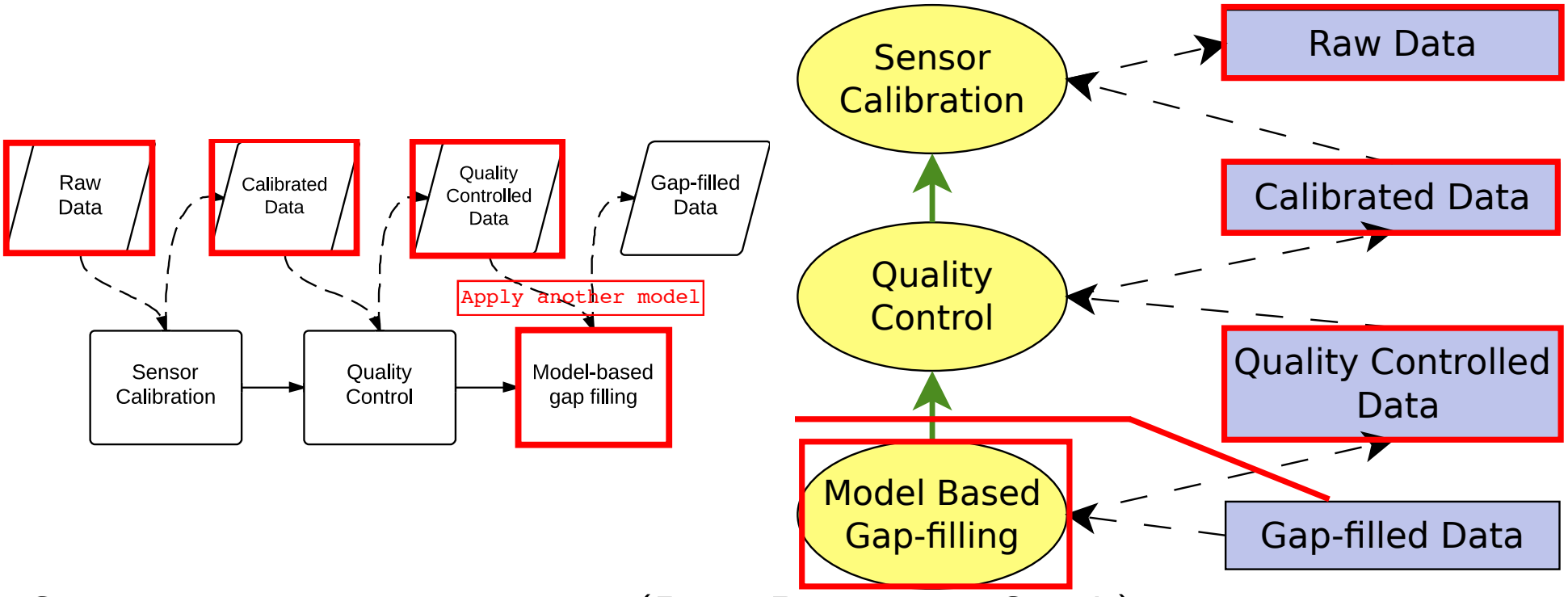




Complete process provenance (Data Derivation Graph)

- Automatically records detailed process execution history
  - data creations and modifications
  - step execution sequences
- Extracts process state at any given point

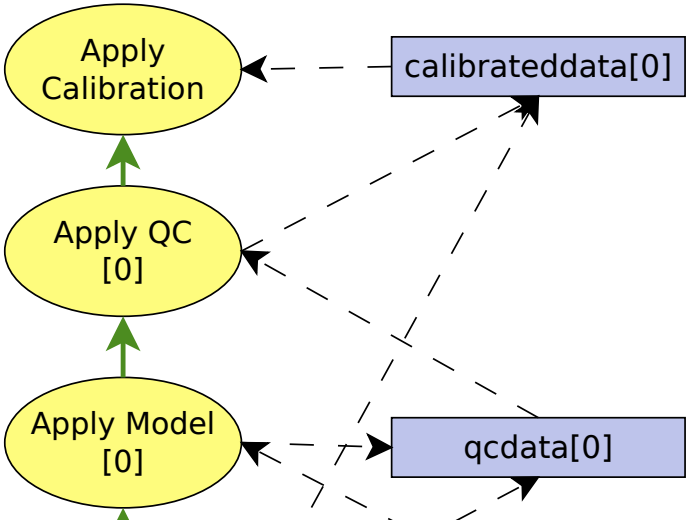
DDG: Provenance Support for Undo/Redo



Complete process provenance (Data Derivation Graph)

- Automatically records detailed process execution history
  - data creations and modifications
  - step execution sequences
- Extracts process state at any given point
- **Undo**: The provenance overrides the current state with the retrieved state, and drives the process.

Using the DDG to Undo



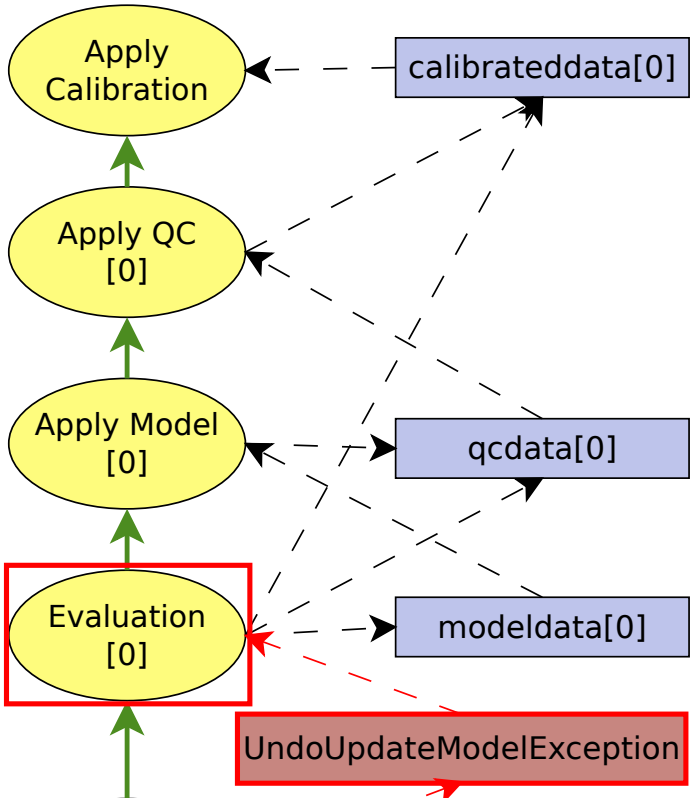
The Scenario

- The scientist decides to apply another model.

Our system will

- present the user with a visualization of the DDG.

Using the DDG to Undo



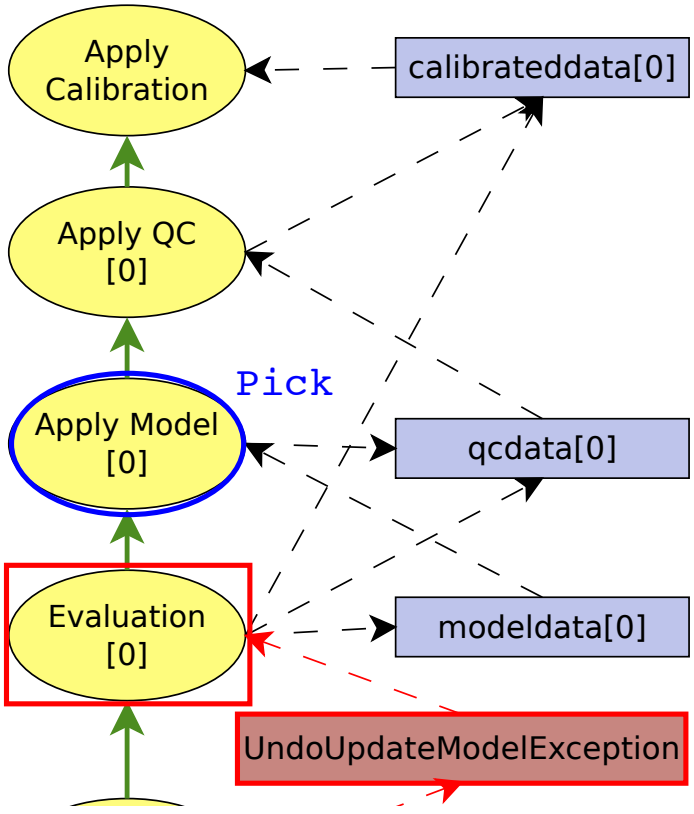
The Scenario

- The scientist decides to apply another model.

Our system will

- present the user with a visualization of the DDG.

Using the DDG to Undo



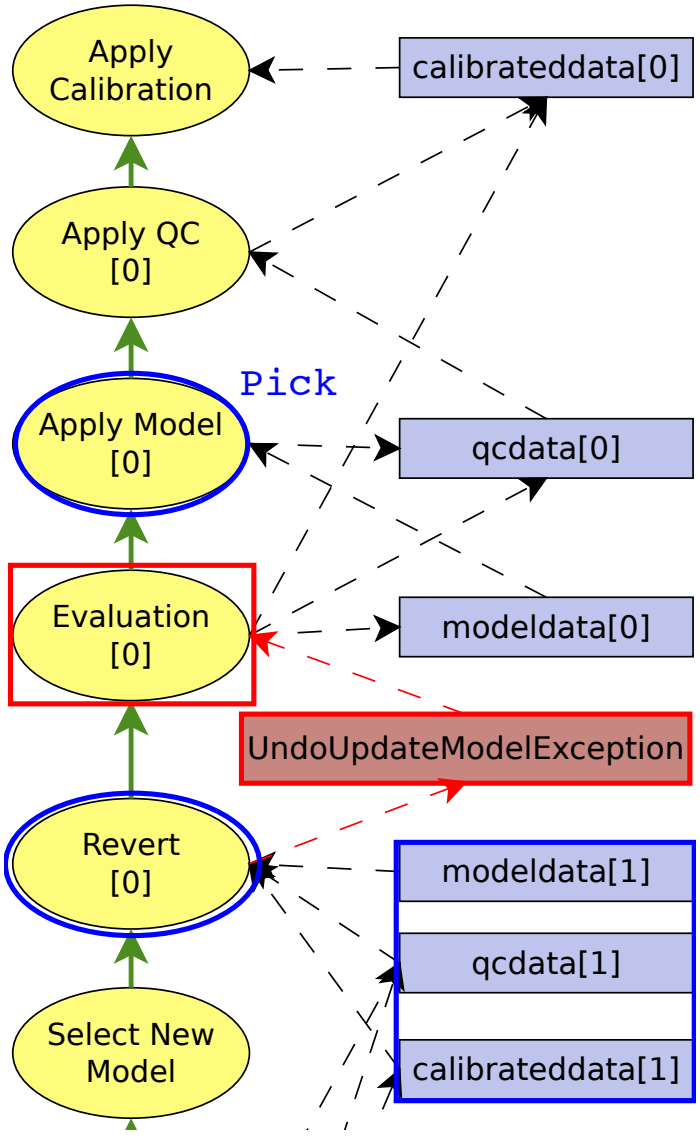
The Scenario

- The scientist decides to apply another model.

Our system will

- present the user with a visualization of the DDG.

Using the DDG to Undo



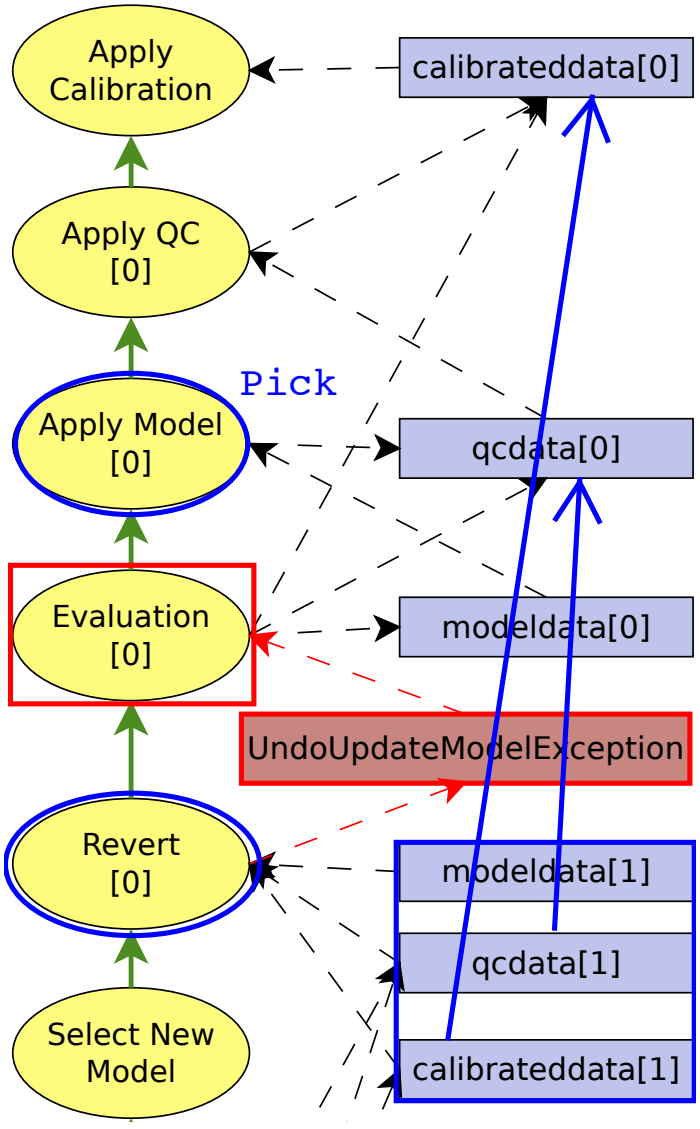
The Scenario

- The scientist decides to apply another model.

Our system will

- present the user with a visualization of the DDG.
- retrieve the appropriate execution state the scientist picks

Using the DDG to Undo



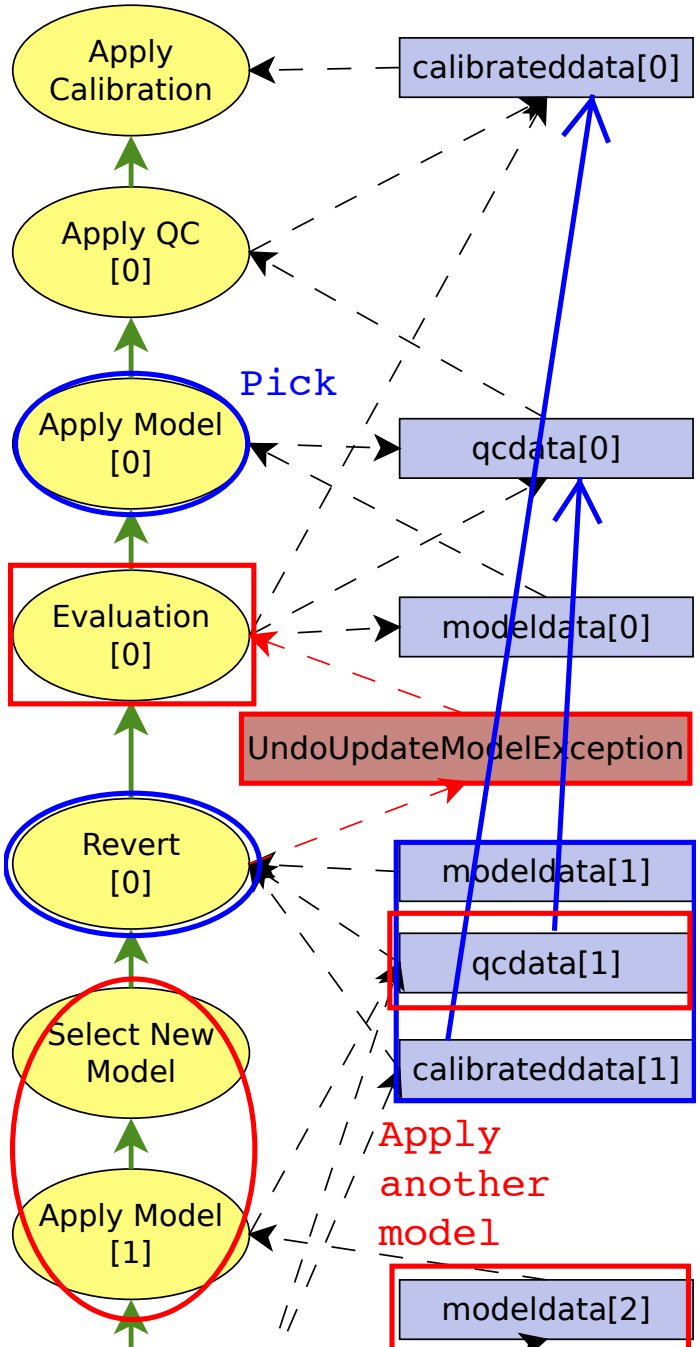
The Scenario

- The scientist decides to apply another model.

Our system will

- present the user with a visualization of the DDG
- retrieve the appropriate execution state the scientist picks

Using the DDG to Undo



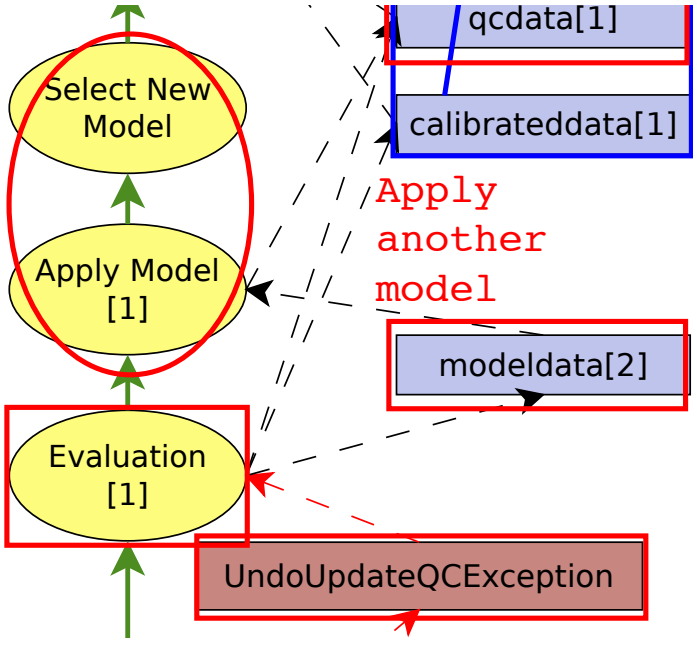
The Scenario

- The scientist decides to apply another model.

Our system will

- present the user with a visualization of the DDG
- retrieve the appropriate execution state the scientist picks
- output the execution state vector and override the current state of the process.

Using the DDG to Undo



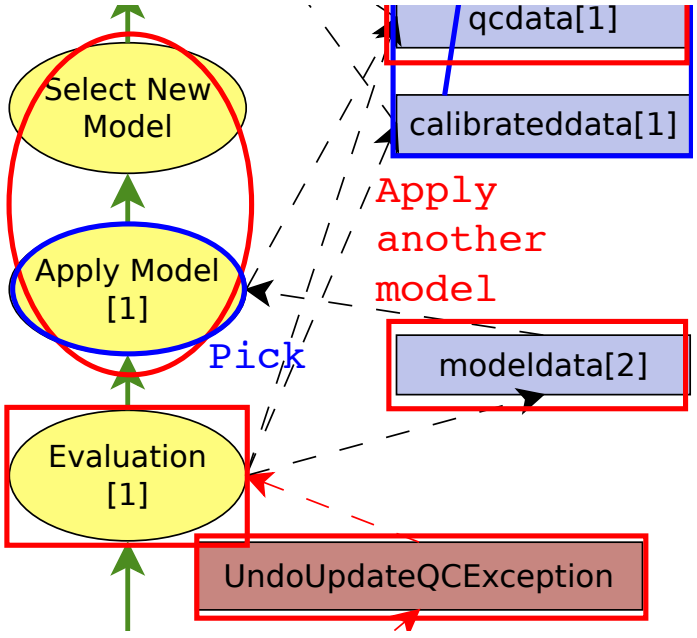
The Scenario

- The scientist decides to apply another model.
- New model applied, evaluation suggests the quality control procedure needs to be reverted.

Our system will

- present the user with a visualization of the DDG
- retrieve the appropriate execution state the scientist picks
- output the execution state vector and override the current state of the process.

Using the DDG to Undo



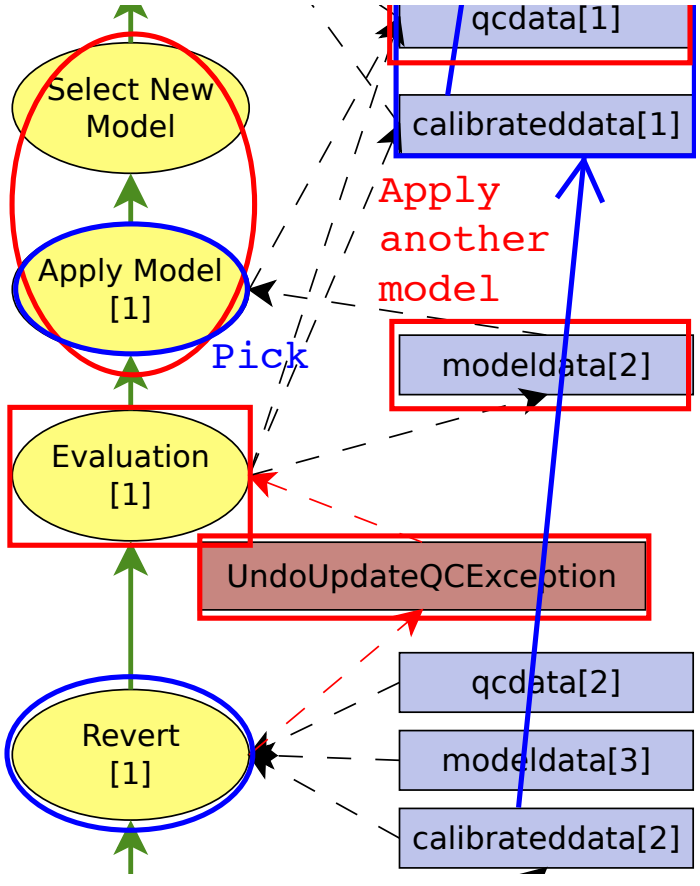
The Scenario

- The scientist decides to apply another model.
- New model applied, evaluation suggests the quality control procedure needs to be reverted.

Our system will

- present the user with a visualization of the DDG
- retrieve the appropriate execution state the scientist picks
- output the execution state vector and override the current state of the process.

Using the DDG to Undo



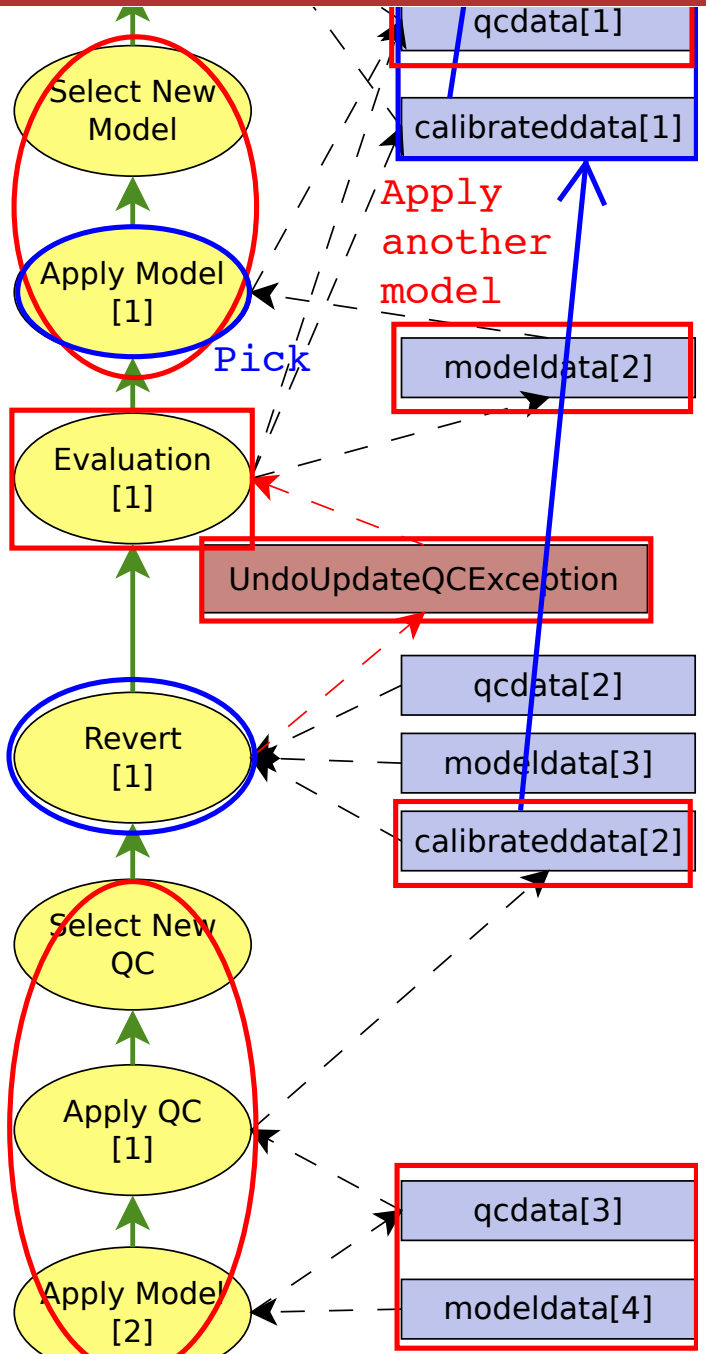
The Scenario

- The scientist decides to apply another model.
- New model applied, evaluation suggests the quality control procedure needs to be reverted.

Our system will

- present the user with a visualization of the DDG
- retrieve the appropriate execution state the scientist picks
- output the execution state vector and override the current state of the process.

Using the DDG to Undo



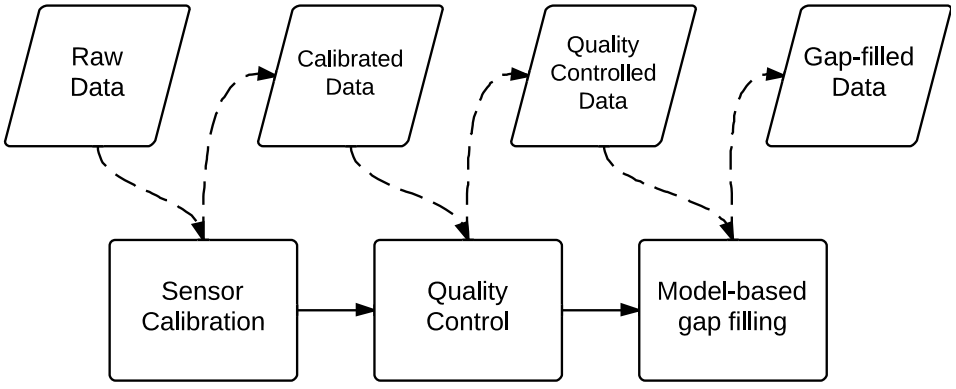
The Scenario

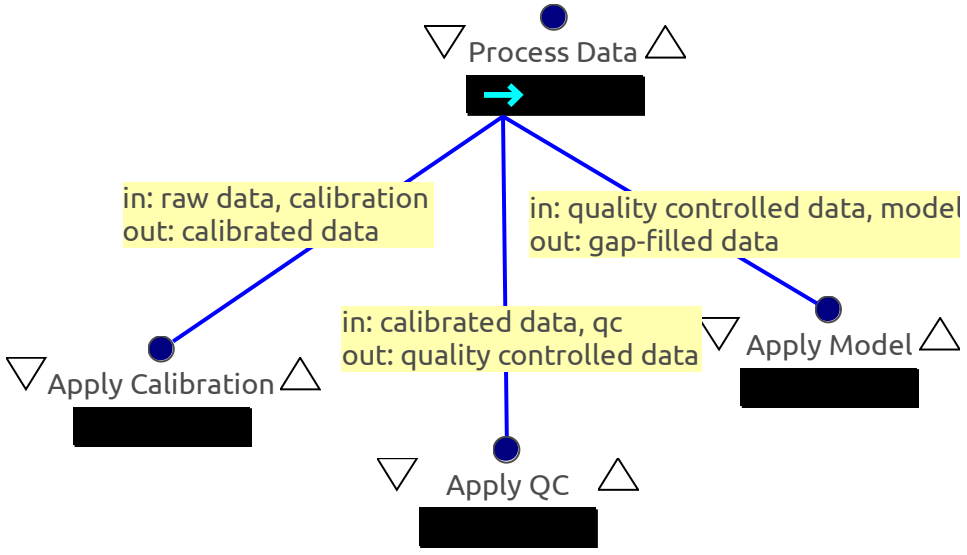
- The scientist decides to apply another model.
- New model applied, evaluation suggests the quality control procedure needs to be reverted.

Our system will

- present the user with a visualization of the DDG
- retrieve the appropriate execution state the scientist picks
- output the execution state vector and override the current state of the process.

Process Support for Undo/Redo

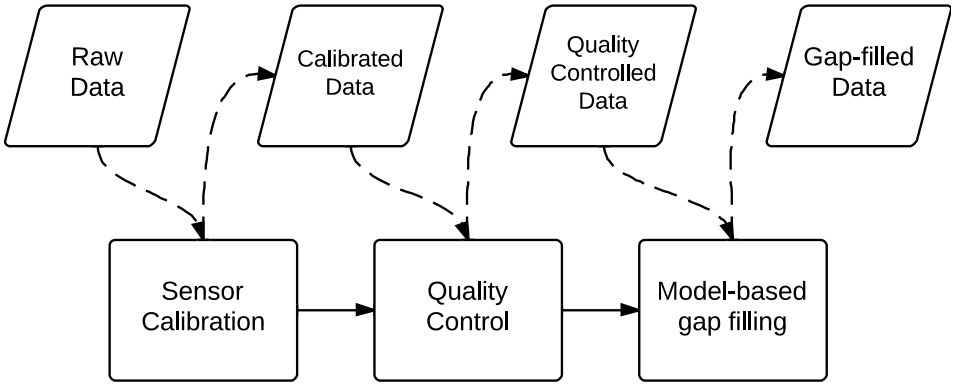


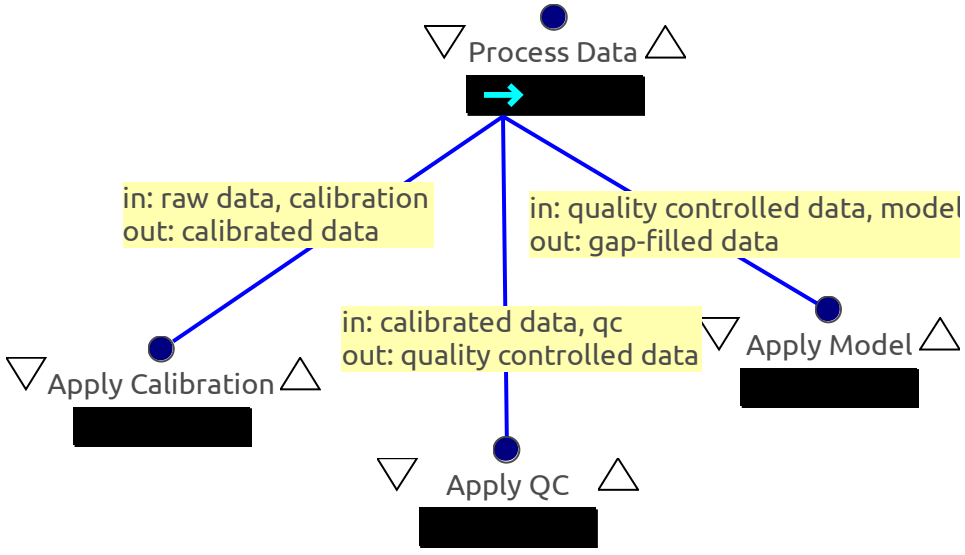


A detailed model of the process (using Little-JIL)

- guides the scientist in undoing and redoing previously executed work in the new context
- allows for tracking & examining the history as the scientist executes it
- manages dataflow and control flow in undo and redo
  - **Undo**: Identify a previously executed step and invoke `Revert`
  - **Redo**: Restore artifact values to previously executed step's values

Process Support for Undo/Redo



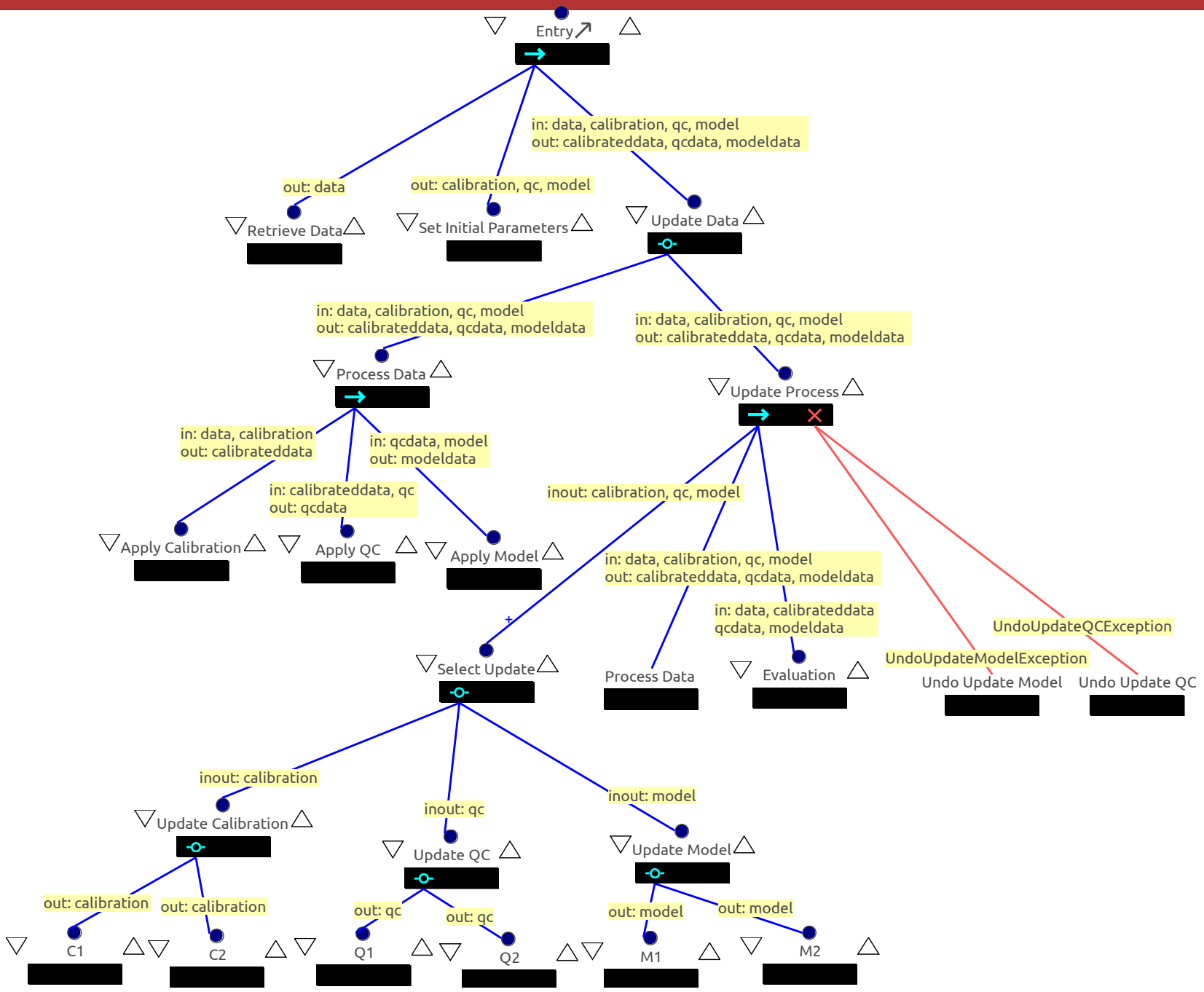


A detailed model of the process (using Little-JIL)

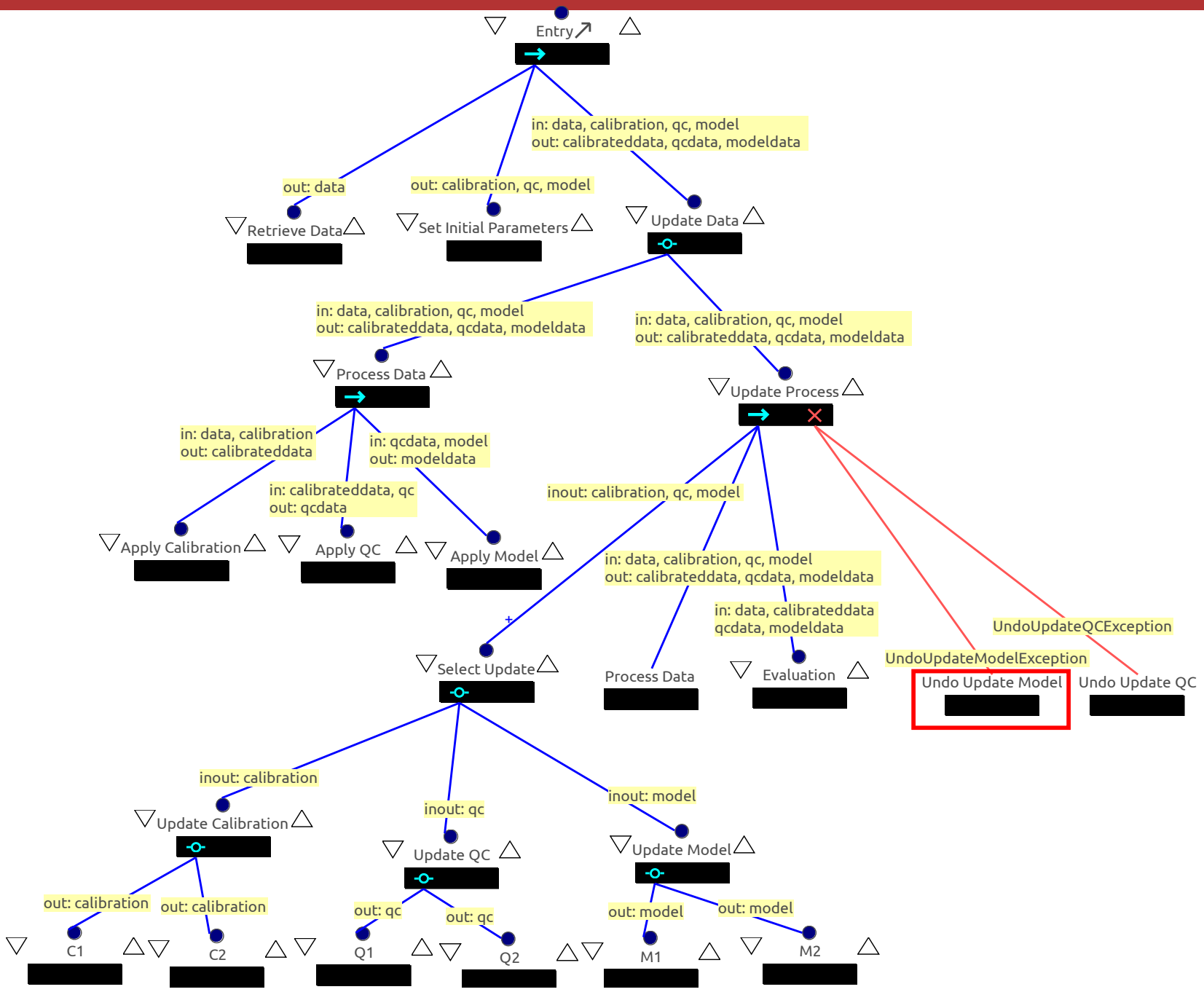
- guides the scientist in undoing and redoing previously executed work in the new context
- allows for tracking & examining the history as the scientist executes it
- manages dataflow and control flow in undo and redo
  - **Undo**: Identify a previously executed step and invoke `Revert`
  - **Redo**: Restore artifact values to previously executed step's values

The scientist needs to design the process beforehand

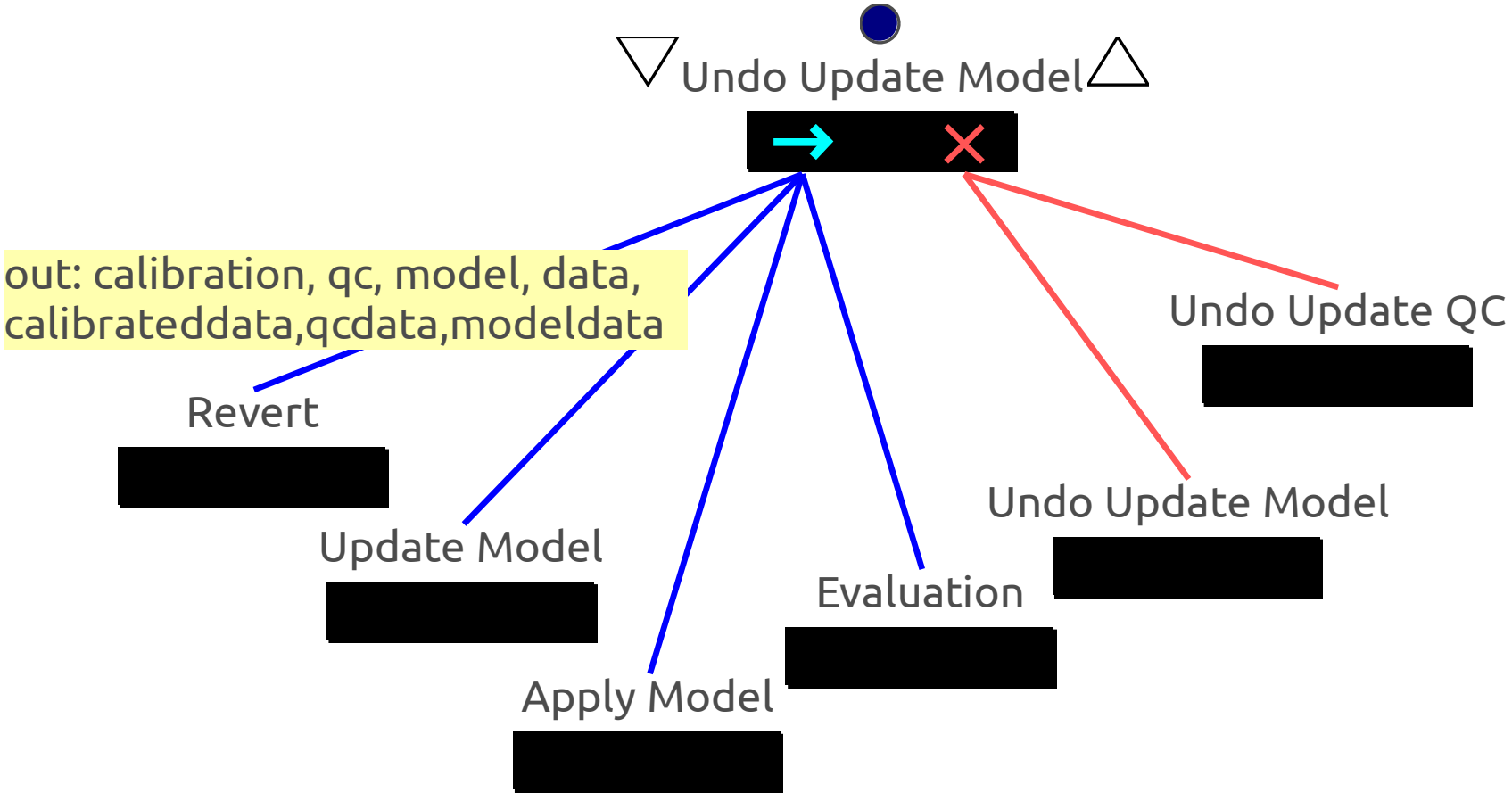
Complete Scientific Data Processing Process Definition



Complete Scientific Data Processing Process Definition

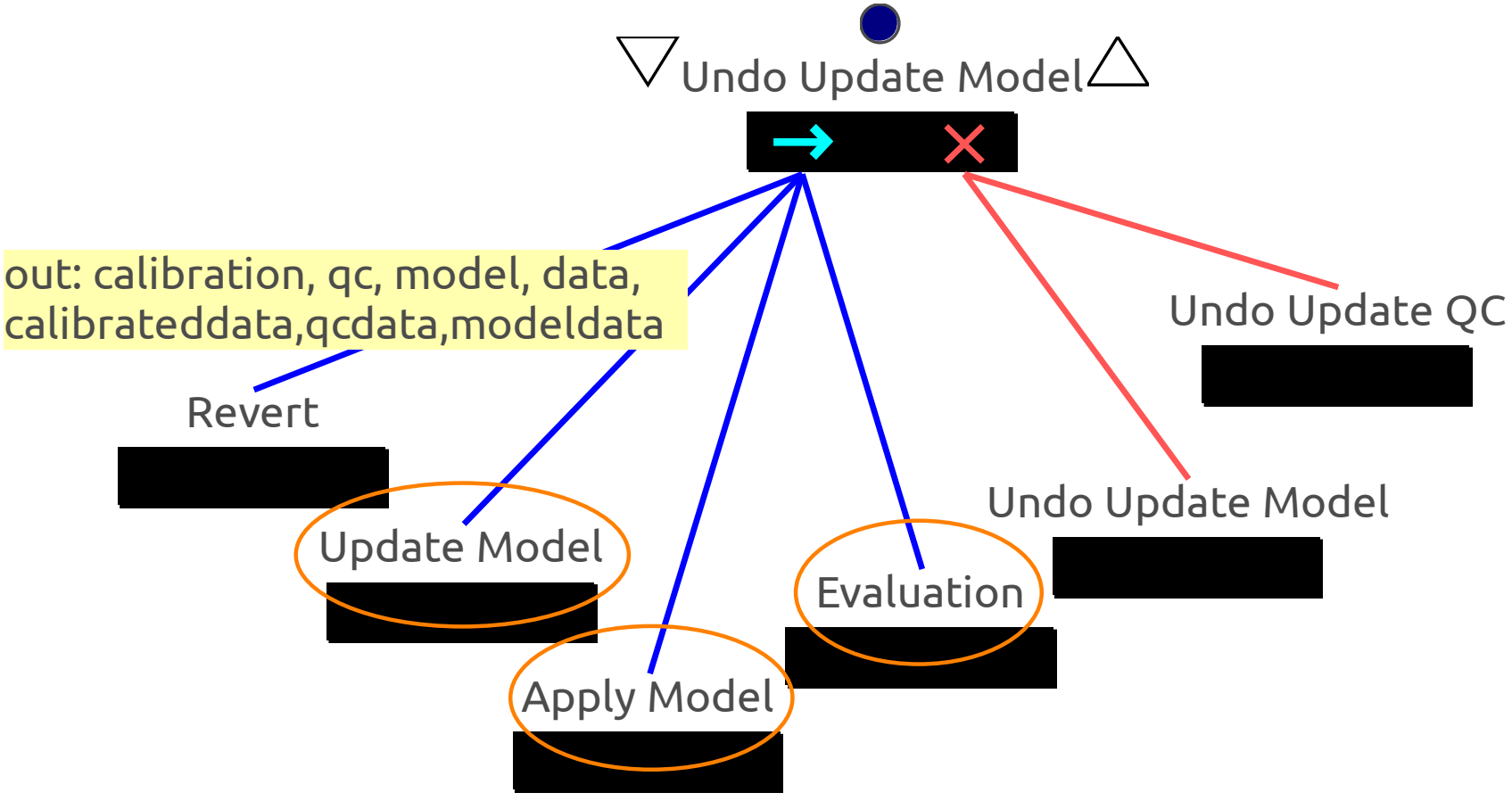


Undo Update Model Step Elaboration



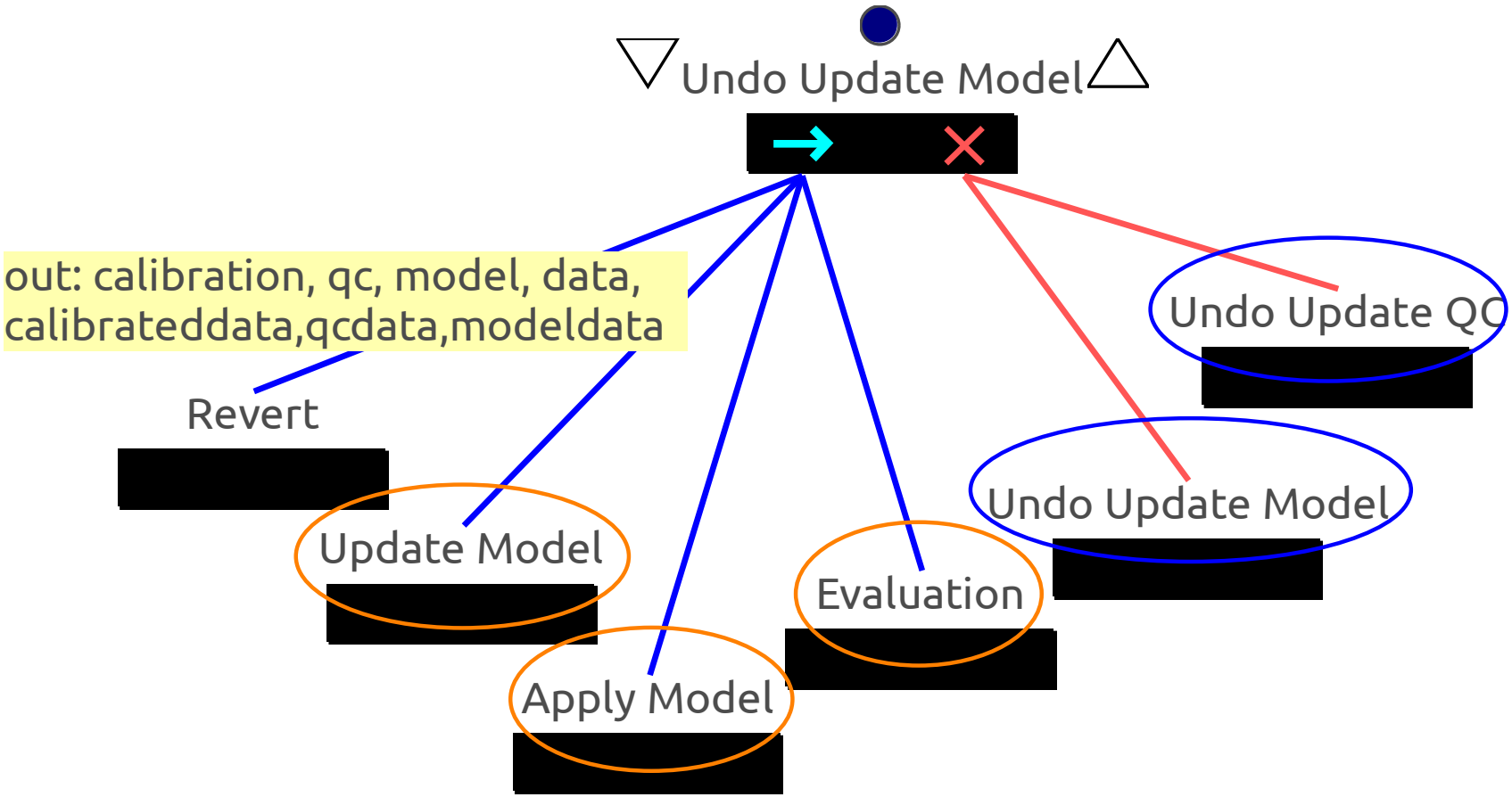
- `Revert` step retrieves the execution state vector at a selected point.

Undo Update Model Step Elaboration



- `Revert` step retrieves the execution state vector at a selected point.
- `Update Model` step is redone, followed by another `Evaluation`.

Undo Update Model Step Elaboration

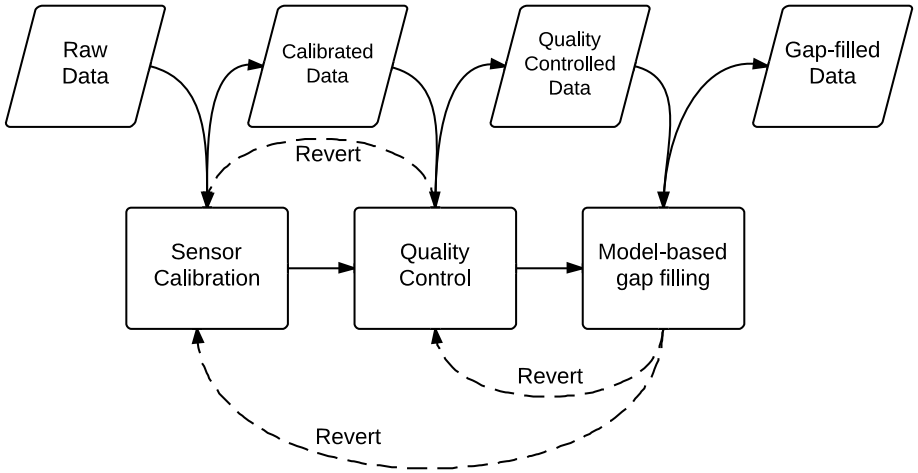


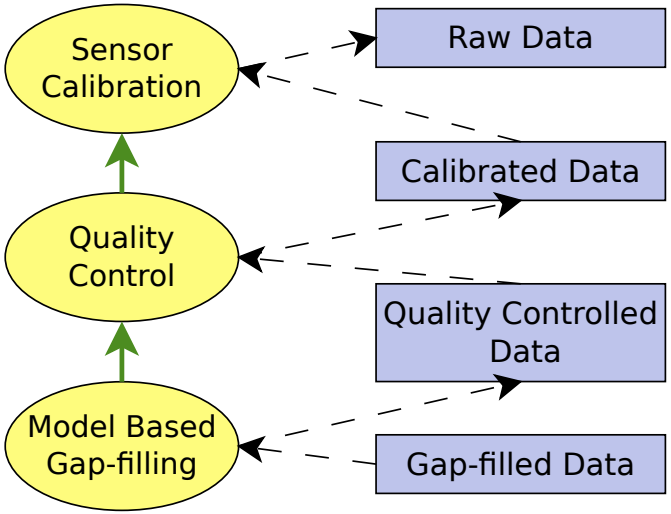
- `Revert` step retrieves the execution state vector at a selected point.
- `Update Model` step is redone, followed by another `Evaluation`.
- Exception handlers can be recursive to assist repetitive undo/redo.

Related Work

- Provenance Visualization
  - Provenance Map Orbiter[Seltzer et al. TaPP '11] captures large provenance graphs and provides navigation mechanism.
  - Navigation model for scientific provenance[Anand et al. WORKS '09].
  - *DDG takes advantage of Little-JIL's hierarchical structure*
- Undo Mechanism
  - [Leeman TPLS '86] proposed a formal approach to undo operations.
  - Selective undo model [Berlage TCHI '94] provides the user with the ability to undo an arbitrary operation in history.
  - *Our approach takes into account both control flow and data flow*
- Undo in WFMSs
  - Kepler tolerates faults by providing check-pointing and forward recovery [Mouallem et al. SSDBM '10].
  - Self-healing Kepler (periodically constructing checkpoints) [Hary et al. HPDC '10].
  - *Our approach is complementary and allows undoing work and trying a different strategy when the results are unsatisfactory*

Contributions and Future Work





Contributions:

- Undo tasks while remembering old artifacts and consequences
- Modify a data-processing step without losing the history
- Automatically redo set-aside tasks that are consistent with the modification

Our approach is implemented as a command-line tool.

Future Work:

- User interface for browsing and querying the DDG
- Detect conflicts in redo operations