Data Model Evolution and Versioning

Klaus Hartke   Ericsson Research   2019-12-17

Versioning Needs

Authors

- "I have made a change to version X. The result is version Y."

- "Version Y broke a customer. We need to roll back to version X."

Implementers

- "We have implemented version X."

- "We have mostly implemented version Y, but some parts are still on version X."

Machines

- "I understand version X."   →   "I understand feature A."

- "I can offer version X and Y."   →   "I can offer feature A and B."

Evolution Ruleset

1. There SHALL be a repository of models.

2. The repository MAY undergo changes.

3. At any time, it SHALL be possible to download a named snapshot of the repository in a known-good state.

4. The name of a snapshot SHALL be a marketing version number for a release and a time stamp or commit hash for a prerelease.

5. Each model SHALL have an identifier that is unique within the repository.

6. The identifier of a model SHALL be opaque (i.e., have no machine-readable semantics).

7. The only possible changes to a repository SHALL be the creation, update, and deprecation of models (i.e., no deletion).

8. Any potential update to a model SHALL be classifiable as either
   - *breaking* (implementation changes are REQUIRED to conform after the update),
   - *non-breaking* (implementation changes are OPTIONAL to conform after the update), or
   - *trivial* (implementation changes are not possible).

9. The only allowed updates to a model SHALL be updates classified as *non-breaking* or *trivial*.