

# Merge-enabled Change Tracking (MCT): Extension prototype

---

Peter Rakyta, MultiRacio Ltd.

Supported by: GOP-1.1.1-11-2011-0006

# Why developing it as an extension?

---

- The code of the extension is separated from the code lines of the OpenOffice.
- Easier development while the specification of MCT is still under construction.
- The same functionality can be adopted in various OpenOffice based applications.

# The main structure of the Extension

---

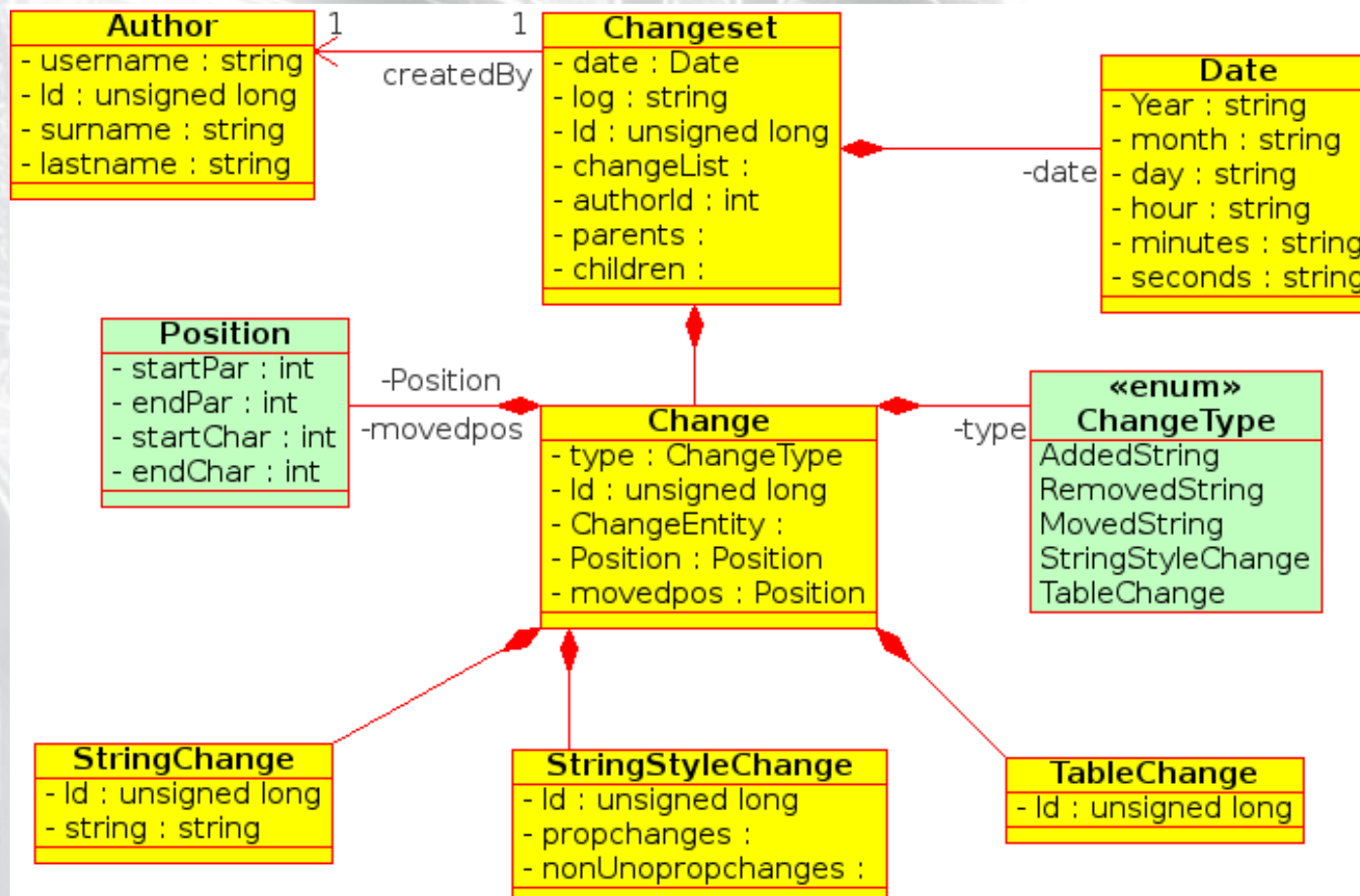
## 1. The model

- Changes stored in a redo/undo graph.
- Should obey the same specification in all implementation.
- XML parser to load and save the history of changes.
- Changes are grouped under changesets (revisions of the document).
- Storing other metadata of the changesets (author, date, comment).

## 2. Connection of the model to the ODF

- Established by the UNO Accessibility API.
- Change tracking based on event listener interfaces provided by the UNO.
- Dialog window in order to control the functionality of the extension.

# The Graph of change history



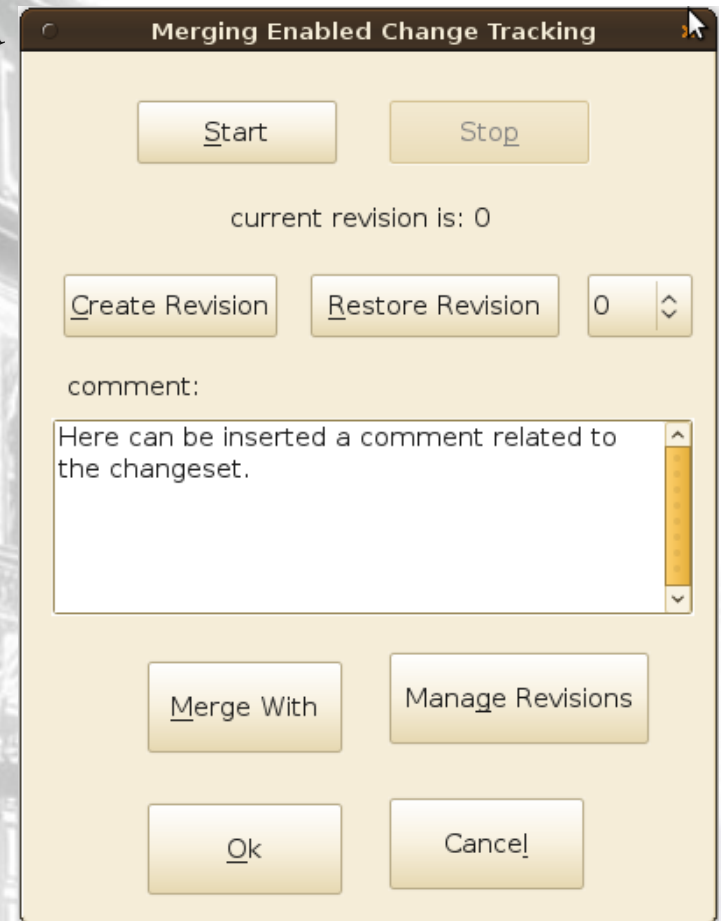
# Currently supported change types

---

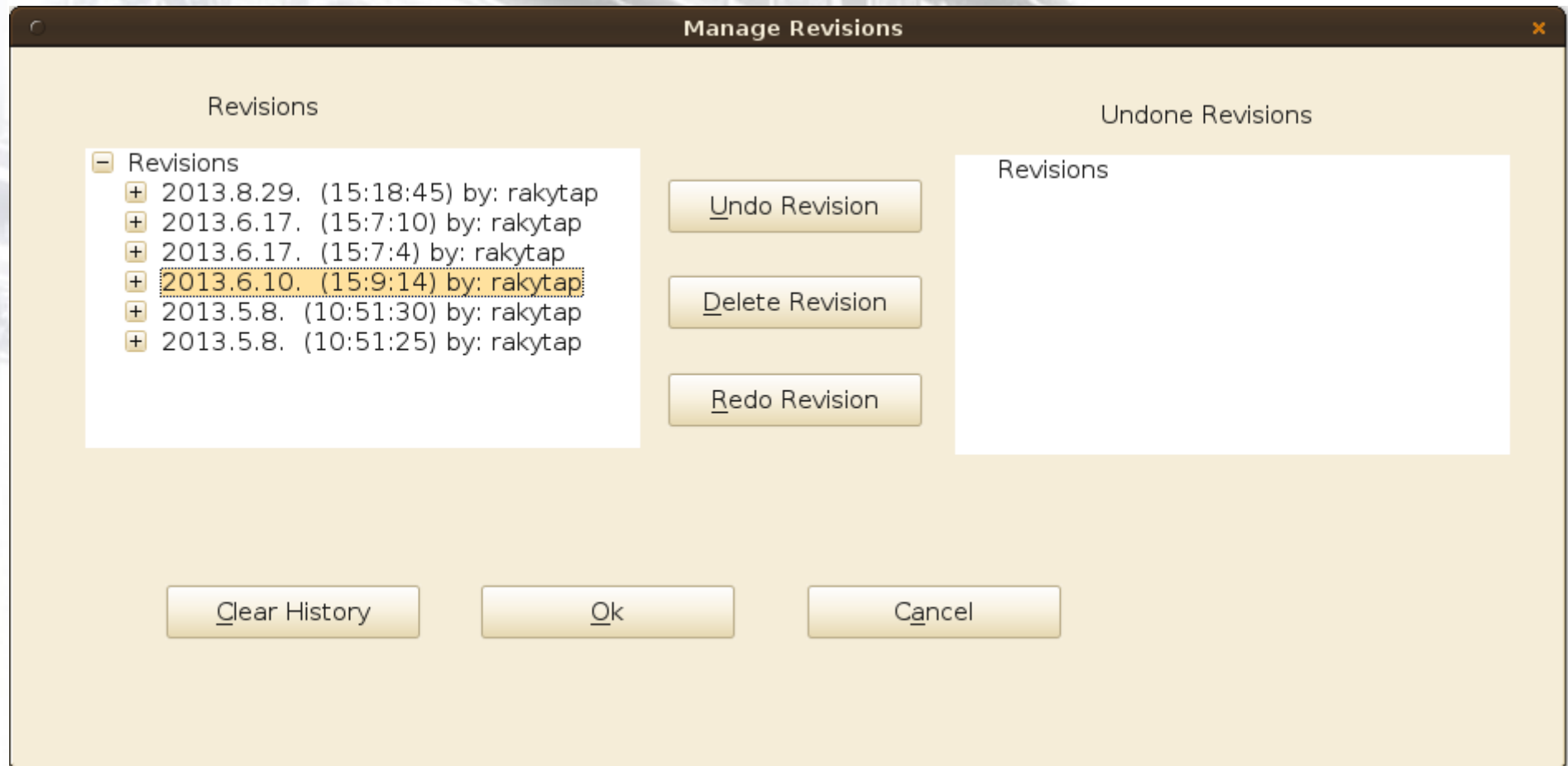
- Recorded changes can be triggered by keyboard or by menu (mouse) events.
- The extension supports Linux and Windows operating systems. (supporting office suits utilizing Python 2.7 and higher is under development)
- Insertion/deletion of characters
- Insertion/deletion of paragraph breaks
- Style change of characters, paragraphs, ordered and unordered lists
- Insertion/deletion of graphic objects

# Creating revisions

- MCT support for a doc is started/stopped by pressing *Start/Stop* button.
- Create revision by *Create Revision* button. (Author and date of the changeset is obtained from OS.)
- Restore document to an earlier state by *Restore Revision* button.
- Manage revisions by *Manage Revisions* button.
- Open merge dialog window by *Merge With* button



# Managing revisions



# Dependency tree of changesets

- Changeset A: insert picture *test.png*
- Changeset B: delete picture *test.png*
- Changeset A **cannot be undone** unless Changeset B is also undone.

The screenshot shows a software interface with two main panels: "Revisions" on the left and "Undone Revisions" on the right. The "Revisions" panel contains a list of revisions with the following details:

- 2013.8.29. (15:18:45) by: rakytap
- 2013.6.17. (15:7:10) by: rakytap
- 2013.6.17. (15:7:4) by: rakytap
- 2013.6.10. (15:9:14) by: rakytap
- 2013.5.8. (10:51:30) by: rakytap
- 2013.5.8. (10:51:25) by: rakytap

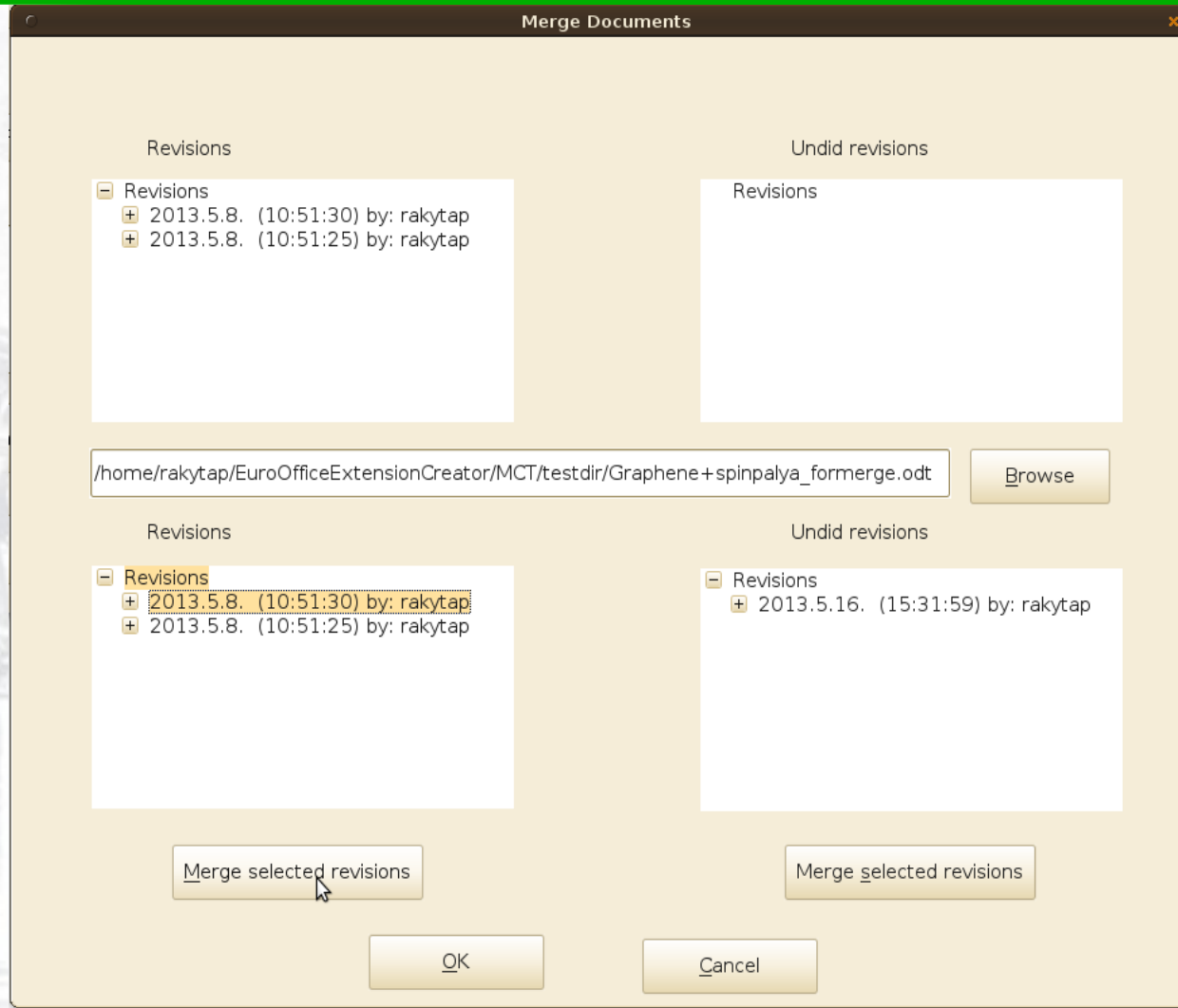
Buttons for "Undo Revision" and "Delete Revision" are visible. A "warning" dialog box is overlaid on the interface, containing the following text:

The following dependent changesets need to be undone in addition:  
2013.6.17. (15:7:4) by: rakytap  
2013.6.17. (15:7:10) by: rakytap

Buttons for "OK" and "Cancel" are present in the dialog. At the bottom of the interface, there are buttons for "Clear History", "Ok", and "Cancel".



# Merging changesets



# Further developments

---

- Extend the list of the supported changetypes:
  - Other types of embedded objects
  - Tables
  - Footnotes
  - etc.
- Developpe a “*tool*” to visualize stored changes:
  - Necessary to understand the changes of collaborators.
  - Necessary to manage revisions.
- Change tracking in spreadsheets.