Compensation in Collaborative Editing

Stéphane Weiss, Pascal Urso, Pascal Molli
{weiss,urso,molli}@loria.fr

Université Henri Poincaré - Nancy I

4 November 2007
Plan

1. Introduction
   - Collaborative editing
   - The Operational Transformation approach (OT)
   - Undoing in the OT approach

2. The compensation approach
   - Presentation
   - Compensation and Integration algorithms
   - Applying the compensation approach

3. Conclusions
Collaborative editing systems

- Modification of shared data can be processed by several users,
- simultaneous editions of a same data (text, XML, image, directory)
- Users are distributed in space and time
- Each site:
  - modifies a copy,
  - generates operations: locally executed, broadcasted, received, re-executed.
Group Undo

- Adaptation of the undo feature from single editors to collaborative editors
- Main issue: how to allow any user to undo any edit operation at anytime?
Introduction
The compensation approach
Conclusions

Collaborative editing
The Operational Transformation approach (OT)
Undoing in the OT approach

Group undo example

Revision history of United States

from Wikipedia, the free encyclopedia
Revision history
View logs for this page

Latest | Earliest View (previous 50) (next 50) (20 | 50 | 100 | 250 | 500)
For any version listed below, click on its date to view it. For more help, see Help:Page history and Help:Edit summary.
(cur) = difference from current version, (last) = difference from preceding version, m = minor edit, → = section edit, ← = automatic edit summary

Compare selected versions

| (last) | 23 Oct 2007 Evb-wiki (Talk | contribs) | (170,156 bytes) | Nwv | [undo] revision 167725909 by Ausrule1921 (talk) | (undo) |
| (last) | 23 Oct 2007 Ausrule1921 (Talk | contribs) | (170,198 bytes) | (undo) |
| (last) | 21 Oct 2007 DCGest (Talk | contribs) | (163,298 bytes) | (undo) |
| (last) | 20 Oct 2007 MSp891 (Talk | contribs) | (163,327 bytes) | (~Demographics) | (undo) |
| (last) | 20 Oct 2007 MSp891 (Talk | contribs) | (163,295 bytes) | (~Demographics) | (undo) |
| (last) | 17 Oct 2007 Evb-wiki (Talk | contribs) | (163,298 bytes) | (undo) |
| (last) | 17 Oct 2007 | (undo) |
| (last) | 14 Oct 2007 Tpark1111 (Talk | contribs) | (163,298 bytes) | (undo) |
| (last) | 14 Oct 2007 Tpark1111 (Talk | contribs) | (132,631 bytes) | (undo) |
| (last) | 14 Oct 2007 ClueBot (Talk | contribs) | (163,298 bytes) | (undo) |
| (last) | 14 Oct 2007 ClueBot (Talk | contribs) | (132,631 bytes) | (~Economy) | (undo) |
Group undo example

"The edit could not be undone due to conflicting intermediate edits."
Plan

1. Introduction
   - Collaborative editing
   - The Operational Transformation approach (OT)
   - Undoing in the OT approach

2. The compensation approach
   - Presentation
   - Compensation and Integration algorithms
   - Applying the compensation approach

3. Conclusions
The Operational Transformation approach (OT)

- Framework for distributed real-time and asynchronous collaborative editors,
- Composed of:
  - Operations,
  - Transformation functions,
  - Integration algorithm.
Transformation functions

Site1

"Compnsation"

\[ \text{Ins}(4, e') \]

"Compensation"

\[ \text{Ins}(11, s') \]

"Compnsations"

Site2

"Compnsation"

\[ \text{Ins}(11, s') \]

"Compensation"

\[ \text{Ins}(4, e') \]

"Compnsations"

\[ \text{Ins}(4, e') \]

"Compensationn"
Introduction
The compensation approach
Conclusions

Collaborative editing
The Operational Transformation approach (OT)
Undoing in the OT approach

Transformation functions

Site1
"Compensation"

Ins(4, 'e')

"Compensation" T "Compensations"

Ins(12, 's')

Site2
"Compensation"

Ins(11, 's')

"Compensations" T "Compensations"

Ins(4, 'e')

"Compensations"
Plan

1. Introduction
   - Collaborative editing
   - The Operational Transformation approach (OT)
   - Undoing in the OT approach

2. The compensation approach
   - Presentation
   - Compensation and Integration algorithms
   - Applying the compensation approach

3. Conclusions
The Anyundo approach

- Defined for the GOTO integration algorithm,
- Treat undo operation specifically,
- Transformation functions must ensure 3 properties.

*Undo any operation at any time in group editors*  – Chengzheng Sun – CSCW 00
The Anyundo approach

- Defined for the GOTO integration algorithm,
- Treat undo operation specifically,
- Transformation functions must ensure 3 properties.

Unfortunately:

- Non-trivial transformation functions satisfying required properties have never been published,

*Undo any operation at any time in group editors*  – Chengzheng Sun – CSCW 00
The undo operation approach

- Applied to the SOCT2 integration algorithm,
- Defines an operation $undo(op)$,
- Transformation functions must ensure 3 properties.

Concurrent Undo Operations in Collaborative Environments Using Operational Transformation – Jean Ferrié and Nicolas Vidot and Michèle Cart – CoopIS/DOA/ODBASE
The undo operation approach

Applied to the SOCT2 integration algorithm,
Defines an operation \textit{undo}(op),
Transformation functions must ensure 3 properties.

Unfortunately:
Cannot ensure consistency.

Concurrent Undo Operations in Collaborative Environments Using Operational Transformation – Jean Ferrié and Nicolas Vidot and Michèle Cart – CoopIS/DOA/ODBASE
COT-UNDO

- Defined for the COT integration algorithm,
- To each operation, a Context Vector is associated,
- Specific mechanism to handle undo op,
- Claim to require only TP1!

*Operation context and context-based operational transformation* – David Sun and Chengzheng Sun – CSCW06 Banff, Alberta, Canada
Gérald Oster’s counter-example

\[
\begin{align*}
&\text{site 1} \quad \text{site 2} \quad \text{site 3} \quad \text{site 4} \\
&\text{"abc"} \quad \text{"abc"} \quad \text{"abc"} \quad \text{"abc"}
\end{align*}
\]

\[
op_1 = \text{del}(2) \quad \nop_2 = \text{ins}(2, x) \quad \nop_3 = \text{ins}(3, y)
\]

\[
\begin{align*}
&ac \\
&axbc \\
&\text{abyyc}
\end{align*}
\]

\[
\begin{align*}
&\text{op}'_2 = \text{ins}(2, x) \quad \text{op}_3 = \text{ins}(3, y)
\end{align*}
\]

\[
\begin{align*}
&\text{op}'_1 = \text{del}(3) \quad \text{op}''_1 = \text{del}(2)
\end{align*}
\]

\[
\begin{align*}
&\text{axyyc} \\
&\text{ayyc}
\end{align*}
\]

\[
\begin{align*}
&\text{op}''_2 = \text{ins}(3, x)
\end{align*}
\]

\[
\begin{align*}
&\text{ayxc}
\end{align*}
\]
Undo counter-example

\[
\begin{align*}
\text{site 1} & \quad \text{"abc"} \\
\text{site 2} & \quad \text{"abc"}
\end{align*}
\]

\[
\begin{align*}
op_1 &= \text{Ins}(2, X, sid_1) \\
op_2 &= \text{Ins}(3, Y, sid_2) \\
op_3 &= \overline{op_1} = \text{Del}(2, sid_1) \\
op_2 &= \text{Ins}(3, Y, sid_2) \\
op_3' &= \overline{op_3'} = \text{Del}(2, sid_1)
\end{align*}
\]

Figure: Using TTF functions with COT.
Finally, the COT approach does not work,
- Context vector’s size grows linearly with the number of operations undone → does not scale.
### Comparisons of all “undo” approach

<table>
<thead>
<tr>
<th></th>
<th>Anyundo</th>
<th>COT-Undo</th>
<th>Undo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consistency</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Instantiable</td>
<td>No</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Properties</td>
<td>TP1, TP2, IP1</td>
<td>TP1</td>
<td>TP1, TP2, IP1</td>
</tr>
<tr>
<td>Integration algorithm</td>
<td>GOTO</td>
<td>COT</td>
<td>SOCT2</td>
</tr>
<tr>
<td>Scalability</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Conclusion

- Undo operations are treated specifically,
- None of the undo approach permits to build a collaborative editor with undo features,
- Need a novel undo approach.
Plan

1. Introduction
   - Collaborative editing
   - The Operational Transformation approach (OT)
   - Undoing in the OT approach

2. The compensation approach
   - Presentation
   - Compensation and Integration algorithms
   - Applying the compensation approach

3. Conclusions
Compensation

- New approach for undo in OT,
- Adaptation of the transactional compensation into the OT approach,
- Undo from the user’s point of view an operation,

**Compensation Property (CP)**

\[ S \circ \text{op} \circ C(\text{op}) \equiv S \]

C(op) is an operation which semantically undoes the effect of the operation op.
Applying the compensation

- For each operation $op$: define a new operation $C(op)$ in a way to satisfy CP,
- Write all transformation functions for all operations (normal + compensation operations)
- Define the function $C()$ which associates to an operation its corresponding compensation operation.
Example

\[
\text{site 1} \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad \\
\quad \quad \quad 
\text{op}_1 = \text{Insert}(0, 'B') \\
\text{"B"}
Example

\[ op_1 = \text{Insert}(0, 'B') \]

\[ op_2 = \text{Insert}(O, 'A') \]

"AB"

"B"

site 1

""

""

S. Weiss

Compensation in Collaborative Editing
Example

\[
\begin{align*}
\text{site 1} & \\
\quad & \\
\text{op}_1 &= \text{Insert}(0, 'B') \\
\quad & \\
\quad & 'B' \\
\text{op}_2 &= \text{Insert}(0, 'A') \\
\quad & \\
\quad & 'AB' \\
\text{op}_3 &= \text{Insert}(2, 'C') \\
\quad & \\
\quad & 'ABC'
\end{align*}
\]
Example

\[ op_4 = C(op_1) = \text{Del}(0) \]

\[ op_1 = \text{Insert}(0, 'B') \]

\[ op_2 = \text{Insert}(0, 'A') \]

\[ op_3 = \text{Insert}(2, 'C') \]
Example

\[ op_4 = C(op_1) = Del(0) \]

\[ op'_4 = Del(1) \]

\[ op_1 = Insert(0, 'B') \]

\[ op_2 = Insert(0, 'A') \]

\[ op_3 = Insert(2, 'C') \]

\[ site 1 \]

\[ "B" \]

\[ "AB" \]

\[ "ABC" \]
Example

\[ op_4 = C(op_1) = Del(0) \]

\[ op_4' = Del(1) \]
Similar to an undo property,

Ensures that anytime you undo an operation, you obtain the same result.
Plan

1. Introduction
   - Collaborative editing
   - The Operational Transformation approach (OT)
   - Undoing in the OT approach

2. The compensation approach
   - Presentation
   - Compensation and Integration algorithms
   - Applying the compensation approach

3. Conclusions
Integration algorithms without undo features

- Compensation does not require special treatment for undo operation → not related to an integration algorithm!
- Integration algorithms required transformation functions satisfying several properties,

⇒ Use SPIKE to write transformation functions satisfying compensating property and integration algorithm properties.
Compensation does not require special treatment for undo operation → not related to an integration algorithm!

Integration algorithms required transformation functions satisfying several properties,

=> Use SPIKE to write transformation functions satisfying compensating property and integration algorithm properties.

Compensation can be applied to all integration algorithms.
Applying compensation on the TTF Functions

- Applied compensation on the TTF functions (Insert and Delete),
- Add compensating operation Undelete to compensate Delete,
- Obtained a set of transformation functions which can be used with all integration algorithms.
Build a prototype using TTF functions extended by compensation and the SOCT2,

Real-time or asynchronous collaborative text editor supporting undo feature.
Conclusions and future work

- Compensation approach is the only “undo” approach instantiable,
- Can be used with all integration algorithms,
- Less restrictive than undo approaches (CP vs IP1),
- Compensation was applied with TTF approach and implemented in a prototype.
- Future work:
  - Implement all integration algorithms in Graveyard,
  - Apply the compensation approach on non-linear structure such as tree.