Uncertain Version Control in Open Collaborative Editing of Tree-Structured Documents

M. Lamine BA, Talel Abdessalem & Pierre Senellart

http://dbweb.enst.fr/
Version Control of Uncertain Data

- Data in large-scale, open and collaborative editing platforms, such as Wikipedia, are inherently uncertain
  - contributors with different reliability, conflicts, malicious edits, ...  
  - need version control to maintain the quality of document versions

- Existing version control approaches are all deterministic
  - no room to uncertainty handling in the versioning process

☞ A version control model aware of uncertain data may be helpful
Uncertain Tree-Structured Data

Unordered, unranked, and labeled XML trees with annotated edges

- annotations are propositional formulas of random Boolean variables

\[
\begin{align*}
\mathcal{P} & \\
\mathcal{e}_1 \lor \mathcal{e}_2 & \quad \mathcal{r} \\
\mathcal{p}_1 & \\
\mathcal{p}_2 & \\
\mathcal{t}_1 & \\
\mathcal{t}_2 & \\
\text{PrXML}^\mathcal{f} & \text{p-Document}
\end{align*}
\]

\[
\begin{align*}
\text{Pr}(\mathcal{e}_1) &= 0.2 \\
\text{Pr}(\mathcal{e}_2) &= 0.8 \\
\text{Pr}(\mathcal{d}_1) &= \text{Pr}(\mathcal{e}_1) \times \text{Pr}(\neg \mathcal{e}_2) \\
\text{Pr}(\mathcal{d}_2) &= (\text{Pr}(\neg \mathcal{e}_1) \times \text{Pr}(\mathcal{e}_2)) + (\text{Pr}(\mathcal{e}_1) \times \text{Pr}(\mathcal{e}_2))
\end{align*}
\]

Possible worlds and their probabilities
Uncertain Tree-Structured Data
Probabilistic XML [Kimelfeld & Senellart.(2013)]

- Unordered, unranked, and labeled **XML trees** with annotated edges
  - annotations are **propositional formulas** of random Boolean variables

\[
\begin{array}{c}
\text{PrXML}^\text{Wie } \text{p-Document} \\
\end{array}
\]

\[
\begin{array}{c}
\text{Possible worlds and their probabilities} \\
\end{array}
\]

\[
\begin{array}{c}
\begin{array}{c}
\text{Pr(e}_1\text{)} = 0.2 \\
\text{Pr(e}_2\text{)} = 0.8 \\
\text{Pr(d}_1\text{)} = 0.04 \\
\text{Pr(d}_2\text{)} = 0.80 \\
\end{array}
\end{array}
\]

\[
\begin{array}{c}
\begin{array}{c}
F_{11} = \{e_1\} \\
F_{21} = \{e_2\} \\
F_{12} = \{e_1, e_2\} \\
F_{22} = \{e_1, e_2\} \\
\end{array}
\end{array}
\]
Uncertain Tree-Structured Data

Probabilistic XML [Kimelfeld & Senellart.(2013)]

- Unordered, unranked, and labeled XML trees with annotated edges
  - annotations are propositional formulas of random Boolean variables

\[
\begin{align*}
\mathcal{P} & \quad r \\
\top & \quad \neg e_2 \\
p_1 & \quad p_2 \\
t_1 & \quad t_2 \\
\text{PrXML} & \quad \text{p-Document}
\end{align*}
\]

- Enumerating all possible worlds and their probabilities

- Enable also to model uncertain updates on (uncertain) nodes [Kharlamov et al.(2010)]

\[
\begin{align*}
\mathcal{F}_{11} & = \{e_1\} \\
\mathcal{F}_{21} & = \{e_2\} \\
\mathcal{F}_{22} & = \{e_1, e_2\}
\end{align*}
\]

Possible worlds and their probabilities
Plan

Uncertain Multi-Version XML Document
Uncertain Version Control Model
Semantics of Updates

Evaluation of the model
Performance Analysis
Filtering capabilities
Uncertain Multi-Version XML Document

Uncertain Version Control Model

Defines two equivalent views over any uncertain multi-version XML tree

- set $V$ of random variables $e_0, e_1 \ldots e_n$ modeling the tree states
- infinite set $\mathcal{D}$ of all (unordered) XML trees including the versions

$\mathcal{G} (\mathcal{G}, \Omega)$: Logical View

- DAG $\mathcal{G}$ built on variables in $V$
- Mapping $\Omega : 2^V \setminus \{e_0\} \rightarrow \mathcal{D}$ which computes the possible versions according to sets of valid events

$\widehat{\mathcal{P}} (\mathcal{G}, \widehat{\mathcal{P}})$: Probabilistic XML Encoding

- Similar DAG $\mathcal{G}$ of random variables in $V$
- Probabilistic XML tree $\widehat{\mathcal{P}}$ which defines the same probability distribution as $\Omega$ mapping
Uncertain Multi-Version XML Document

Uncertain Version Control Model (Example)

\[ d_1 \) person \[
\quad \text{name} \]
\quad “Obama”
\quad \mathcal{F}_1 = \{ e_1 \}

\[ e_2 \]
\quad \text{name} \rightarrow \text{origin}
\quad “Obama” “Tanzania”
\quad \mathcal{F}_2 = \{ e_1, e_2 \}

\[ e_3 \]
\quad \text{name} \rightarrow \text{origin}
\quad “B. Obama” “Kenya”
\quad \mathcal{F}_3 = \{ e_1, e_3 \}

\[ d_2 \)
\quad \text{person} \quad \text{origin}
\quad “Obama” “Tanzania”

\[ d_3 \)
\quad \text{person} \quad \text{origin}
\quad “B. Obama” “Kenya”

\[ d_4 \)
\quad \text{person} \quad \text{function}
\quad “Obama” “Tanzania” “US. President”
\quad \mathcal{F}_4 = \{ e_1, e_2, e_4 \}
Uncertain Multi-Version XML Document

Uncertain Version Control Model (Example)

\[ G \]
\[ e_0 \rightarrow e_1 \]
\[ e_1 \rightarrow e_2 \]
\[ e_1 \rightarrow e_3 \]
\[ e_2 \rightarrow e_4 \]

\[ d_1 \) person
\[ \) name
\[ \) “Obama”
\[ \) \( F_1 = \{ e_1 \} \)

\[ d_2 \) person
\[ \) name
\[ \) “Obama”
\[ \) origin
\[ \) “Tanzania”
\[ \) \( F_2 = \{ e_1, e_2 \} \)

\[ d_3 \) person
\[ \) name
\[ \) “B. Obama”
\[ \) origin
\[ \) “Kenya”
\[ \) \( F_3 = \{ e_1, e_3 \} \)

\[ d_4 \) person
\[ \) name
\[ \) “Obama”
\[ \) origin
\[ \) “Tanzania”
\[ \) function
\[ \) “US. President”
\[ \) \( F_4 = \{ e_1, e_2, e_4 \} \)

\[ d_5 \) person
\[ \) name
\[ \) “B. Obama”
\[ \) origin
\[ \) “Kenya”
\[ \) function
\[ \) “US. President”
\[ \) \( F_5 = \{ e_1, e_3, e_4 \} \)
Uncertain Multi-Version XML Document

Uncertain Version Control Model (Example)

Encode uncertain changes over nodes with formulas on edges

$F_1 = \{ e_1 \}$

$F_3 = \{ e_1, e_3 \}$

$F_5 = \{ e_1, e_3, e_4 \}$

“Obama”

“B. Obama”

“Tanzania”

“Kenya”

“US. President”

$G$

$P$

$\mathcal{G}$

$\mathcal{P}$

$d_1$

$d_2$

$d_3$

$d_4$

$d_5$
Uncertain Multi-Version XML Document

Semantics of Updates

- Assume an event $e_i$ in $G$ pointing to the edited version
- Come with a new event $e_j$ not in $G$ and an edit script $\Delta$

Logical definition

Input: $(G, \Omega), e_i, e_j, \Delta$

- $G := G \cup \{(e_j, (e_i, e_j))\}$
- Extension of $\Omega$ to a $\Omega'$ mapping

For each event set $\mathcal{F} \in 2^{(\mathcal{V}\setminus\{e_0\})\cup\{e_j\}}$:

$\Rightarrow \Omega'(\mathcal{F}) = [\Omega(\mathcal{F}\setminus\{e_j\})]^\Delta$ if $e_j \in \mathcal{F}$

$\Rightarrow \Omega'(\mathcal{F}) = [\Omega(\mathcal{F})] \text{ if } e_j \notin \mathcal{F}$

Probabilistic XML Update

Input: $(G, \hat{P}), e_i, e_j, \Delta$

- $G := G \cup (\{e_j\}, \{(e_i, e_j)\})$
- Updating $\hat{P}$ with operations in $\Delta$

For an insert of $x$ and a delete of $y$:

$\Rightarrow fie(x) := fie(x) \lor (e_j)$ if $x \in \hat{P}$ or insert $x$ in $\hat{P}$ with $fie(x) := (e_j)$

$\Rightarrow fie(y) := fie(y) \land \neg(e_j)$
Plan

Uncertain Multi-Version XML Document
Uncertain Version Control Model
Semantics of Updates

Evaluation of the model
Performance Analysis
Filtering capabilities
Evaluation of the model
Performance Analysis

Estimation of two main metrics: commit time and checkout cost

Baseline Systems

- Versioning tools SubVersion and Git
  - Use of their Java implementations based on the APIs SvnKit and JGit

Real Datasets

- History of commits over two large file systems (shared tree-structured data)
  - Linux kernel development
  - Cassandra project

- Set up our system (PrXML) in Java language

- Measures are obtained with all accesses in RAM Disk
Evaluation of the model
Performance Analysis (Results)

Figure: commit time over real-world datasets (logarithmic y-axis)
Evaluation of the model

Performance Analysis (Results)

Figure: checkout time over real-world datasets (linear axes)
Evaluation of the model
Filtering capabilities

- Tests are run over a sample of articles from the Wikipedia dump
  - Automatic filtering of unreliable content, e.g. spams, in versions of articles
  - Generate arbitrary versions that fit user preference
    - versions from trustworthy authors
  - Test more advanced operations over critical articles such as vandalized pages
    - e.g. study the impact of considering as reliable some versions affected by vandalism in the history of the edition of a given article
  - Detection of vandalism as well as Wikipedia robots do, automatically manage it while keeping all uncertain versions available for checkout.
  - etc.
Evaluation of the model

Filtering capabilities (Demo [Ba et al.(2011)])

(a) keyword-based search engine
(b) generation of arbitrary versions
(c) visualization features

13th ACM Symposium on DocEng – Sept 10-13, Florence (Italy)
Thank for your attention!
References


