Contrôle de version incertain dans l’édition collaborative ouverte de documents arborescents

M. Lamine BA, Talel Abdessalem & Pierre Senellart

BDA’13 – 22-25 October 2013 (Nantes, France)

http://dbweb.enst.fr/
Versioning on the Web is Uncertain (I)

- Large-scale, open and collaborative editing platforms, e.g., wikipedia
  - **Unreliable contributors**, Novice vs. Experts, etc.
  - Malicious edits, Vandalism acts, **Contradictions**
Versioning on the Web is Uncertain (II)

**Version control** is used in large-scale web collaborative editing platforms in order to **integrate** contributions from different sources and to support **fixing errors** with the possibility to query previous **data versions**

- No notion of more **relevant** versions or contributions which will fit to **user-preferences**, but just the concept of last **valid** revisions

- **Deterministic** version control models [Kerstin et al.(2009), Al Koc et al.(2011)] in the literature
Motivations

Versioning on the Web is Uncertain

\[ v_0 (a_0, c_0) \quad \text{derivation} \quad v_1 (a_1, c_1) \quad \text{derivation} \quad v_2 (a_2, c_2) \quad v_4 (a_1, c_4) \]

\[ v_3 (a_3, c_3) \]
Versioning on the Web is Uncertain (III)

\[ v_0 \ (a_0, \ c_0) \quad v_1 \ (a_1, \ c_1) \quad v_2 \ (a_2, \ c_2) \quad v_4 \ (a_1, \ c_4) \]

\[ Q_1 : \neg a_1 \]
\[ Q_2 : \neg c_1 \]
\[ Q_3 : \forall v_i \ | \ Pr(v_i) \neq 0 \]
Motivations

Versioning on the Web is Uncertain (III)

\[ v_0 (a_0, c_0) \]
\[ v_1 (a_1, c_1) \]
\[ v_2 (a_2, c_2) \]
\[ v_3 (a_3, c_3) \]
\[ v_4 (a_1, c_4) \]

\[ Q_1 : v_2 \models \neg a_1 \]
\[ Q_2 : v_2 \models \neg c_1 \]
\[ Q_3 : \text{all } v_i \models \Pr(v_i) \neq 0 \]

Uncertain version control in Open Collaborative Contexts
Versioning on the Web is Uncertain (IV)
Versioning on the Web is Uncertain (IV)

\[ F = \{ \epsilon_1, \epsilon_3, \epsilon_4 \} \]
Versioning on the Web is Uncertain (IV)

\[ P = \{ e_1, e_3, e_4 \} \]
Plan

Uncertain Tree-Structured Data Model

Uncertain Multi-Version XML Document

Conclusion and Further work
Uncertain Tree-Structured Data

Probabilistic XML [Kimelfeld & Senellart.(2013)]

\[
\begin{align*}
\Pr(e_1) &= 0.2; \quad \Pr(e_2) = 0.8 \\
\Pr(d_1) &= \Pr(e_1) \times \Pr(\neg e_2) \\
\Pr(d_2) &= (\Pr(\neg e_1) \times \Pr(e_2)) + (\Pr(e_1) \times \Pr(e_2)) \\
\Pr(d_3) &= \Pr(\neg e_1) \times \Pr(\neg e_2)
\end{align*}
\]
Uncertain Tree-Structured Data

Probabilistic XML [Kimelfeld & Senellart.(2013)]

\[ \widehat{\mathcal{P}} \]

\[ e_1 \lor e_2 \]

\[ \neg e_2 \]

\[ \begin{align*}
& p_1 & p_2 \\
& t_1 & t_2 \\
& v_1 = (e_1 \land \neg e_2) & v_2 = (\neg e_1 \land e_2) & v_3 = (\neg e_1 \land \neg e_2)
\end{align*} \]

\[ \begin{align*}
& d_1) \quad r \\
& d_2) \quad r \\
& d_3) \quad r
\end{align*} \]

\[ \begin{align*}
& Pr(e_1) = 0.2; \quad Pr(e_2) = 0.8 \\
& Pr(d_1) = 0.04 \\
& Pr(d_2) = 0.80 \\
& Pr(d_3) = 0.16
\end{align*} \]
Uncertain Tree-Structured Data
Probabilistic XML [Kimelfeld & Senellart.(2013)]

\[\widehat{\mathcal{P}}\] = \langle \mathcal{T}, C(E), fie, Pr \rangle

\[\widehat{\mathcal{P}}\] = \langle D, Pr \rangle; \quad \sum\{Pr(d) \mid d \in D\} = 1
Plan

Uncertain Tree-Structured Data Model

Uncertain Multi-Version XML Document
Uncertain Version Control Model
Probabilistic XML Encoding
Updating Uncertain Multi-version XML documents
Performance Analysis

Conclusion and Further work
Uncertain Multi-Version XML Document
Uncertain Version Control Model

- **Probability space** (PSV) over Uncertain versions of XML documents
- **Random** derivation graph (DG) over the document versions produced

**Intuition:** states in DG are complex variables $e_i$’s called **version control events** based on simpler variables $b_1 \ldots b_m$ managing uncertainty in data

$< \mathcal{G}, \Omega >$ defines a multi-version XML document with uncertain data

- $\mathcal{G}$ is DG over a set of versioning events $\mathcal{V} \cup e_0$ with $\mathcal{V} = \{e_1 \ldots e_n\}$
- $\Omega : 2^{\mathcal{V}\setminus \{e_0\}} \rightarrow \mathcal{D}$ a mapping computing the PSV according to sets of valid events
Possible versions

- Version control events come with edit scripts updating content

- $\forall i, \forall \mathcal{F} \subseteq 2^\mathcal{V}\setminus\{e_i\}$, the possible version $\Omega(\{e_i\} \cup \mathcal{F}) = [\Omega(\mathcal{F})]^{\Delta_i}$

Probability of possible versions

- Assume a prior probability distribution over simple variables $b_1 \ldots b_m$

- The probability of a given possible version $\Omega(\mathcal{F})$ is the probability of $\forall \mathcal{F} \subseteq \mathcal{V}, \Omega(\mathcal{F}) \mathcal{F}$
Uncertain Multi-Version XML Document

Probabilistic XML Encoding

- **Compact** representation of all possible versions (PSV) in $\Omega$ mapping

Intuition: Represent PSV using **propositional formulas** of simple variables $b_1 \ldots b_m$ attached to nodes in a global tree $T$ containing all provided data

A probabilistic XML encoding of $< G, \Omega >$ is a couple $< G, \hat{P} >$ with

- $\hat{P} = < T, C(V), fie, Pr >$ is a probabilistic XML document

**Thm1:** $[\hat{P}]$ defines the same probability distribution over $D$ as $\Omega$, i.e.,

$$< G, [\hat{P}] > = < G, \Omega >$$
Uncertain editions in $\Delta$ over nodes of a possible tree version

An update is an uncertain version control event defined based on a triple $< e_i, e_j, \Delta >$ ($e_i \in \mathcal{G}$ and $e_j \not\in \mathcal{G}$)

$\text{updOP}(< e_i, e_j, \Delta >)[< \mathcal{G}, \Omega >]$

- $\mathcal{G} := \mathcal{G} \cup \{e_j\}, \{(e_i, e_j)\}$

- Extension of $\Omega$ to a $\Omega'$ mapping with for each $\mathcal{F} \in 2(\mathcal{V} \setminus \{e_0\}) \cup \{e_j\}$
  $\Omega'(\mathcal{F}) = [\Omega(\mathcal{F} \setminus \{e_j\})]^\Delta$ if $e_j \in \mathcal{F}$ and $\Omega'(\mathcal{F}) = \Omega(\mathcal{F})$ otherwise
Uncertain Multi-Version XML Document

Updating Uncertain Versions (II)

\[ \text{updPrXML}(< e_i, e_j, \Delta >) \cdot [< \widehat{P}, \Omega >] \]

- \( G := G \cup \{e_j\}, \{(e_i, e_j)\} \)
- For all \( \text{ins}(x, i) \) in \( \Delta \)
  - Set \((x, \text{fie}(x) \lor (e_j))\) if \( x \) already in \( \widehat{P} \)
  - Insert \( x \) in \( \widehat{P} \) and Set \((x, (e_j))\) otherwise
- For all \( \text{del}(x) \) in \( \Delta \)
  - Set \((x, \text{fie}(x) \land \neg (e_j))\)

**Thm2:** \( \text{updOP}(< e_i, e_j, \Delta >) \equiv \text{updPrXML}(< e_i, e_j, \Delta >) \)

**Thm3:** \( \text{updPrXML} \) runs in \( O(1) \) while \( \widehat{P} \) grows linearly according to \( |\Delta| \)
Uncertain Multi-version XML documents
Performance Analysis (Metrics, datasets and baseline)

- Estimation of the commit time and checkout cost of the model

**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

- Implementation of our system (PrXML) in Java

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

Commit time (ms)

Subversion
Git
PrXML

Commit time (ms)

Subversion
Git
PrXML

Commit (Linux kernel)

Commit (Cassandra project)
Evaluation of the model

Performance Analysis (Checkout time)

Revision (Linux kernel)

Checkout time (ms)

Revision (Cassandra project)

Checkout time (ms)
Plan

Uncertain Tree-Structured Data Model

Uncertain Multi-Version XML Document

Conclusion and Further work
Conclusion and Further work (i)

- Design of a probabilistic version control approach for uncertain tree-structured documents
  - Both logical description and an efficient probabilistic compact XML encoding
  - Set-up of the most used version control operation, i.e., update and with an efficient mapping algorithms
  - Both theoretical and practical complexity analysis of the proposed model

- Extension of our model in [Ba et al.(DChanges, 2013)] with the merging operation over uncertain versions
Conclusion and Further work (ii)

- Support of more complex versioning operations such as copying, renaming etc.

- Study the impact of introducing some constraints over the order of nodes in uncertain version control
MERCI!
References


