Towards a Version Control Model with Uncertain Data

M. Lamine BA *, Talel Abdessalem **, Pierre Senellart **

* Université Cheikh Anta DIOP (Senegal)
** Télécom ParisTech (France)

PhD Students in Information and Knowledge Management, 28 October 2011
Glasgow, Scotland, UK

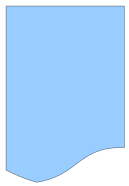
Outline

What is a version control model ?

- Data management approach that maintains at the same time several data versions when updates are performed.

Consider a file in the two following cases

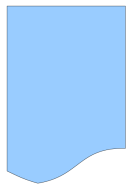
update operations →



file is updated

old changes are lost

non-versioning system



v_0 v_1

two file versions are saved

version control system

Main features

- Not overwriting old changes
- Tracking history of updates
- Set of operations on versions

. . .

· · · subject to multiple interests in numerous domains

- seen as an effective support for data exchange (or sharing) within large communities.

Version Control Supports Systems with Uncertain Data

- In case of data manipulated in a collaborative manner.

Common example is collaborative editing of documents, as in

- Office-based collaborative editing applications or;
- Content-based online collaborative platforms.

Various reasons of uncertainties in data

- Unreliable data sources or inherent uncertain data.
- Data with variable relevance depending on the source.
- Semantic problems leading sometimes to uncertainties.
- And so on. . .

Wikipedia Revisions as pratical case

A web-based platform for content driven collaboration

- An online encyclopedia built on free contributions.
- Promote contributions of everyone and from everywhere.
- Encourage content of high quality provided by experts.

Wikipedia documents as a succession of revisions

- Version control, the heart of the system, manages edits.

Problems in version control are :

- irrevelant content and edit wars
- trustworthiness of data sources

$\Rightarrow$ time-consuming or error-prone tasks

Wikipedia Revisions as pratical case

Data are inherently uncertain

- Incomplete data or unreliable sources.
- Spams or edit wars can introduce of such uncertain data.
- Semantic nature of a conflict, i.e., on what is really true.

How to deal with these uncertainties in Wikipedia ?

Investigate a versioning-driven approach with uncertain data.

Objective :

- integrate uncertainty management in version control
- extend semantics of query on versions to uncertainties
- benefit of reputation and trust algorithms, e.g., [MCA11]

We focus on a XML version control model.

Outline

Uncertainty management in XML

Extensively studied in various application areas, such as:

- Integration of heterogeneous Web data sources.
- Synchronisation of XML Databases in mobile P2P systems.

· · · aims at
reducing tedious and time-consuming human intervention.
modelizing rigorously, with minimun efforts, uncertainties.

many research efforts
- p-models & implemented tools

Uncertainty management in XML

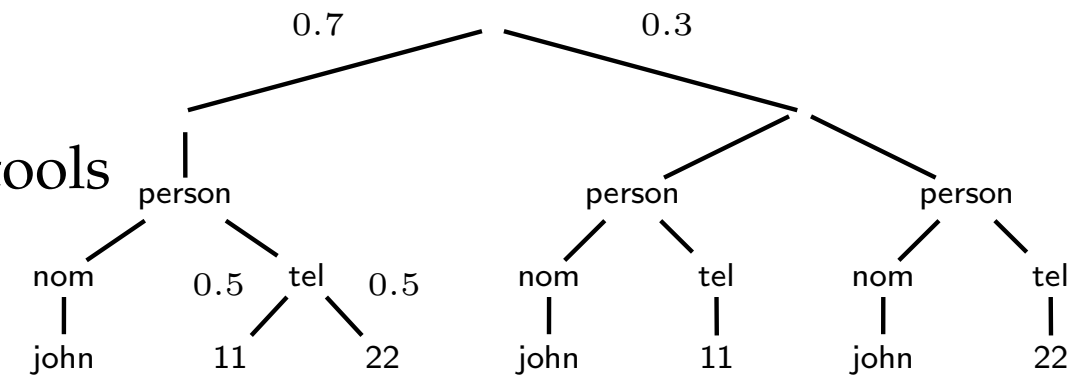
Extensively studied in various application areas, such as:

- Integration of heterogeneous Web data sources.
- Synchronisation of XML Databases in mobile P2P systems.

· · · aims at
reducing tedious and time-consuming human intervention.
modelizing rigorously, with minimun efforts, uncertainties.

many research efforts

- p-models & implemented tools



Model of possible worlds [vKdKA05]

Uncertainty management in XML

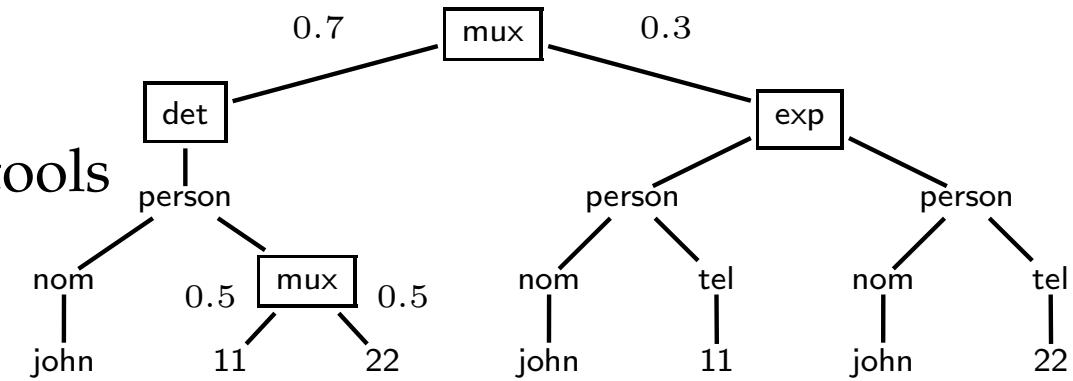
Extensively studied in various application areas, such as:

- Integration of heterogeneous Web data sources.
- Synchronisation of XML Databases in mobile P2P systems.

· · · aims at
reducing tedious and time-consuming human intervention.
modelizing rigorously, with minimun efforts, uncertainties.

many research efforts

- p-models & implemented tools
  - compact and expressive



Model of p-documents [AKSS09, KNS10]

Uncertainty management in XML

Extensively studied in various application areas, such as:

- Integration of heterogeneous Web data sources.
- Synchronisation of XML Databases in mobile P2P systems.

· · · aims at
reducing tedious and time-consuming human intervention.
modelizing rigorously, with minimun efforts, uncertainties.

many research efforts

- p-models & implemented tools
  - IMPrECISE [dKvK08], Prob XML Merging Tool [ABS11], . . .

Uncertainty management in XML

Extensively studied in various application areas, such as:

- Integration of heterogeneous Web data sources.
- Synchronisation of XML Databases in mobile P2P systems.

· · · aims at
reducing tedious and time-consuming human intervention.
modelizing rigorously, with minimun efforts, uncertainties.

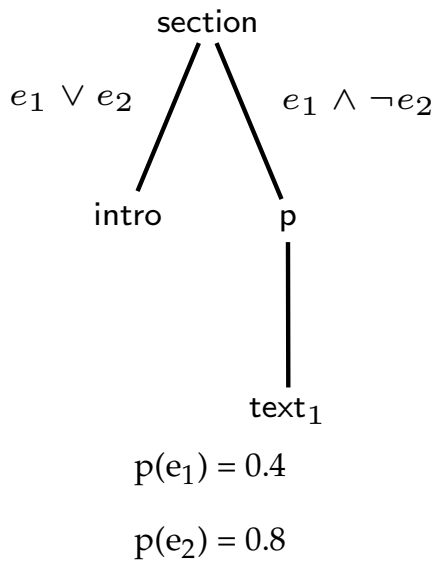
many research efforts

- p-models & implemented tools

uncertainties representation

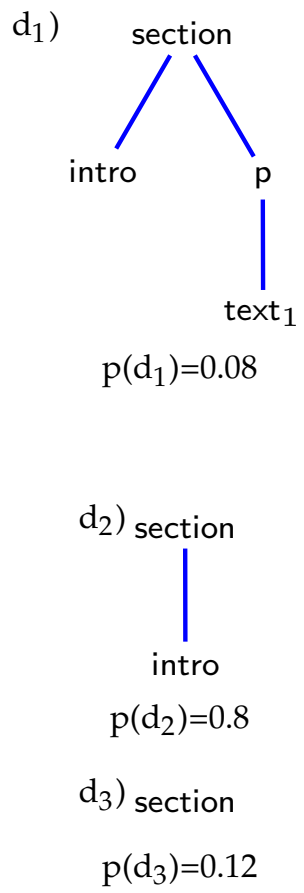
- quantitative or qualitative methods
  - numerical values or random variables

PrXML^{fie} model of P-Documents



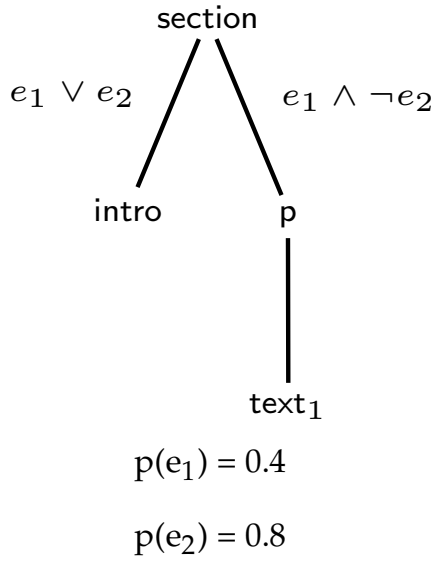
- Ordered XML trees with probabilistic data.
  - prob distribution over XML documents.
  - outcome of prob XML merging or prob XML updates.
- Prob data modeled as formula of events.
  - in trees, represent annotations on edges.
- Events a.k.a random Boolean variables.
  - capture several semantics, e.g. data existence.
  - independent events, i.e. $p(e_i \wedge e_j)=p(e_i)p(e_j), \forall\ e_i \neq e_j$.

PrXML^{fie} model of P-Documents



- Ordered XML trees with probabilistic data.
  - prob distribution over XML documents.
  - outcome of prob XML merging or prob XML updates.
- Prob data modeled as formula of events.
  - in trees, represent annotations on edges.
- Events a.k.a random Boolean variables.
  - capture several semantics, e.g. data existence.
  - independent events, i.e. $p(e_i \wedge e_j)=p(e_i)p(e_j), \forall\ e_i \neq e_j$.
- represent possibilities and their probabilities.

PrXML^{fie} model of P-Documents



- Ordered XML trees with probabilistic data.
  - prob distribution over XML documents.
  - outcome of prob XML merging or prob XML updates.
- Prob data modeled as formula of events.
  - in trees, represent annotations on edges.
- Events a.k.a random Boolean variables.
  - capture several semantics, e.g. data existence.
  - independent events, i.e. $p(e_i \wedge e_j)=p(e_i)p(e_j), \forall\ e_i \neq e_j$.

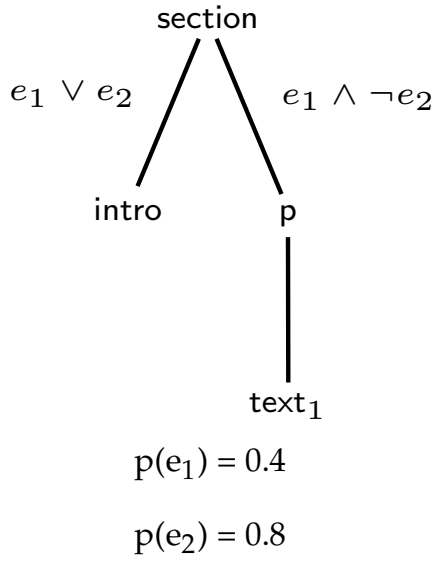
Main results

- Time-efficient for all kind of updates.
- Manage both lineage and uncertainty.
- Probability of all non-trivial queries is costly to compute.

updates (tree-pattern)	PrXML^{fie} model
insertion	L/P
deletion	L/P

Data complexities of updates [KNS10]

PrXML^{fie} model of P-Documents



- Ordered XML trees with probabilistic data.
  - prob distribution over XML documents.
  - outcome of prob XML merging or prob XML updates.
- Prob data modeled as formula of events.
  - in trees, represent annotations on edges.
- Events a.k.a random Boolean variables.
  - capture several semantics, e.g. data existence.
  - independent events, i.e. $p(e_i \wedge e_j)=p(e_i)p(e_j), \forall\, e_i \neq e_j$.

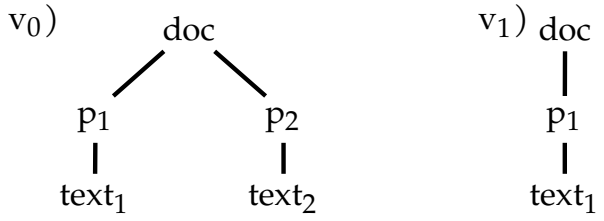
Some useful properties for XML version control

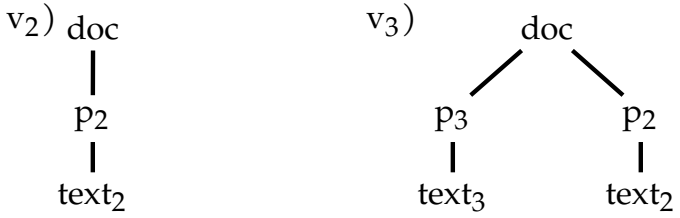
- allow to maintain simultaneously all versions of a document
- ensure basic operations on versions as known in versioning domain

$\cdots$ and beyond that

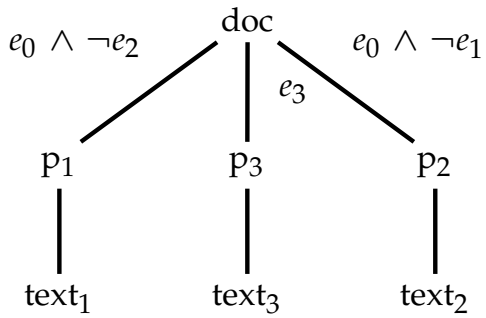
- allow for uncertainty management in XML version control

PrXML^{fie} p-document of versions integration





Example of document versions in a collaborative system



Resulting PrXML^{fie} p-document

Prob XML merging algorithm from [ABS11]

- some modifications in the basic form
  - e.g. do not assume successive versions.
- Each version v_i comes with event e_i.
- Matching of versions v_i and v_j ($i \leq j$)
  - Deleted nodes x : $x \in v_i$ and x has no match in v_j.
  - Added nodes x : $x \in v_j$ and has no match in v_i.
  - Matched couples (x,y) : $x \in v_i$ and $y \in v_j$ match.
- Updating p-document with matches describing semantics of changes.
  - Deleted nodes x : $fie_{new}(x) = fie_{old}(x) \wedge (\neg e_j)$
  - Added nodes x : $fie_{new}(x) = fie_{old}(x) \vee e_j$ or $fie_{new}(x) = e_j$
  - Unmodified nodes x : $fie_{new}(x) = fie_{old}(x)$

Outline

XML differencing algorithm

Main requirement in version control of XML documents

- Detect changes between XML versions.
  - simple or structural XML updates
- Represent semantics of updates.
  - edit operations or transformation script

Time-and space-efficient XML diff algorithms

Based on following aspects

- XML data model and delta model

Some expected performance criteria

- Optimal diff or approximation thereof.
- Possibly, a linear-time algorithm.

- Proposals assume an ordered tree model.
- Produce correct XML diff in reasonable time.
- XyDiff [Mar02], Faxma [LKT06], and XCC [RB10]

Outline

Towards an Uncertain XML Version Control

- Assume a doc $\mathcal{D}$ edited in a collaborative and open manner.
- Propose to manage uncertainty in $v_0, v_1, \ldots, v_n$ versions of $\mathcal{D}$.
  - Assumption: each version v_i is coming with a random variable e_i.

Towards an Uncertain XML Version Control

- Assume a doc $\mathcal{D}$ edited in a collaborative and open manner.
- Propose to manage uncertainty in v_0, v_1,. . ., v_n versions of $\mathcal{D}$.
  - Assumption: each version v_i is coming with a random variable e_i.

Three main points for the intended model

Towards an Uncertain XML Version Control

- Assume a doc $\mathcal{D}$ edited in a collaborative and open manner.
- Propose to manage uncertainty in $v_0, v_1, \ldots, v_n$ versions of $\mathcal{D}$.
  - Assumption: each version v_i is coming with a random variable e_i.

Three main points for the intended model

GRAPH OF VERSION DERIVATIONS $\rightarrow$

Describe a DAG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with

$\mathcal{V}$: a set of events as nodes and;

$\mathcal{E}$: dependencies between nodes.

Deduce a Bayesien Network $\mathcal{B} = (\mathcal{G}, \Theta)$ on $\mathcal{G}$ where $\Theta = \{(v_i, v_j), v_i \Rightarrow v_j \text{ and } p(e_j|e_i)=1\}$

Towards an Uncertain XML Version Control

- Assume a doc $\mathcal{D}$ edited in a collaborative and open manner.
- Propose to manage uncertainty in v_0, v_1,. . ., v_n versions of $\mathcal{D}$.
  - Assumption: each version v_i is coming with a random variable e_i.

Three main points for the intended model

GRAPH OF VERSION DERIVATIONS $\rightarrow$ Describe a DAG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V}$: a set of events as nodes and; $\mathcal{E}$: dependencies between nodes. Deduce a Bayesien Network $\mathcal{B} = (\mathcal{G}, \Theta)$ on $\mathcal{G}$ where $\Theta = \{(v_i,v_j), v_i \Rightarrow v_j$ and $p(e_j|e_i)=1\}$

PRXML^{fie}P-DOCUMENT AT EACH STATE $\rightarrow$ Merge of all versions v_k, for $0 \leq k \leq n$, using an extension of algorithm in [ABS11]. Describe state of versioned document at each version.

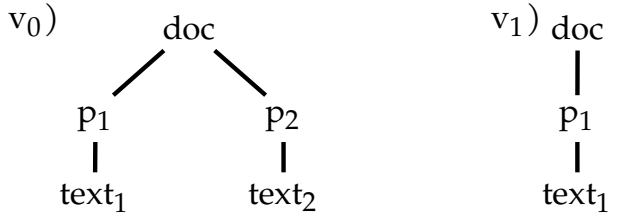
Towards an Uncertain XML Version Control

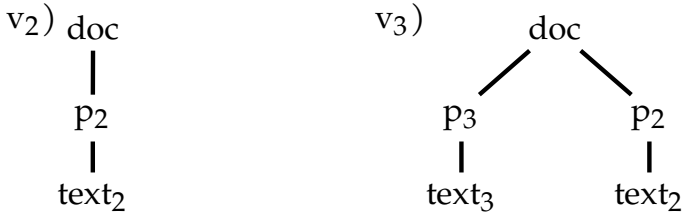
- Assume a doc $\mathcal{D}$ edited in a collaborative and open manner.
- Propose to manage uncertainty in $v_0, v_1, \ldots, v_n$ versions of $\mathcal{D}$.
  - Assumption: each version v_i is coming with a random variable e_i.

Three main points for the intended model

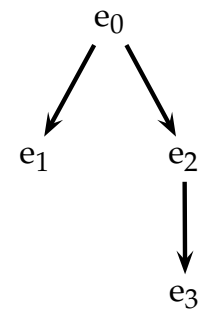
GRAPH OF VERSION DERIVATIONS	→	Describe a DAG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V}$: a set of events as nodes and; $\mathcal{E}$: dependencies between nodes. Deduce a Bayesien Network $\mathcal{B} = (\mathcal{G}, \Theta)$ on $\mathcal{G}$ where $\Theta = \{(v_i, v_j), v_i \Rightarrow v_j$ and $p(e_j \mid e_i)=1\}$
PRXML^{fie} P-DOCUMENT AT EACH STATE	→	Merge of all versions v_k, for $0 \leq k \leq n$, using an extension of algorithm in [ABS11]. Describe state of versioned document at each version.
OPERATIONS OR UPDATES ON THE P-DOC	→	Translate semantics of operations in queries on p-docs. Extend queries to uncertainties.

Example of the versioning process



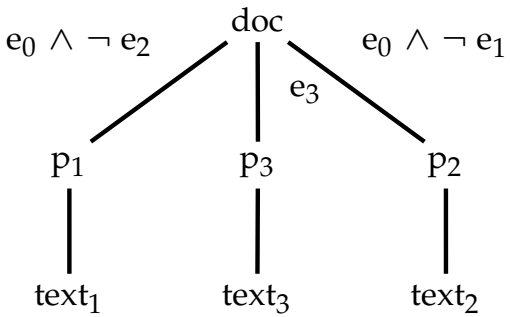


Versions with Uncertain Data



$\mathcal{L} = \{(v_0, v_1), (v_0, v_2), (v_2, v_3)\}$

Graph of version derivations

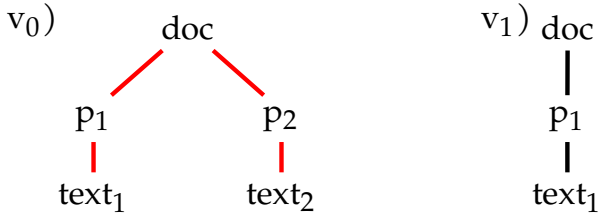


$Pr(e_0)=0.5$; $Pr(e_1)=0.4$

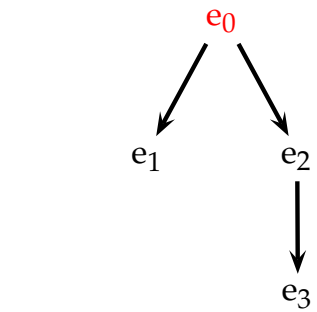
$Pr(e_2)=0.7$; $Pr(e_3)=0.5$

Result of the merge ($\mathcal{P}$)

Example of the versioning process

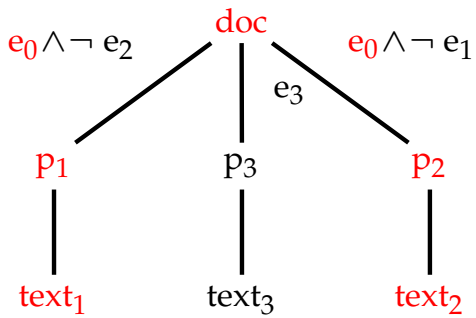


Versions with Uncertain Data



$\mathcal{L} = \{(v_0, v_1), (v_0, v_2), (v_2, v_3)\}$

Graph of version derivations



$Pr(e_0)=0.5$; $Pr(e_1)=0.4$

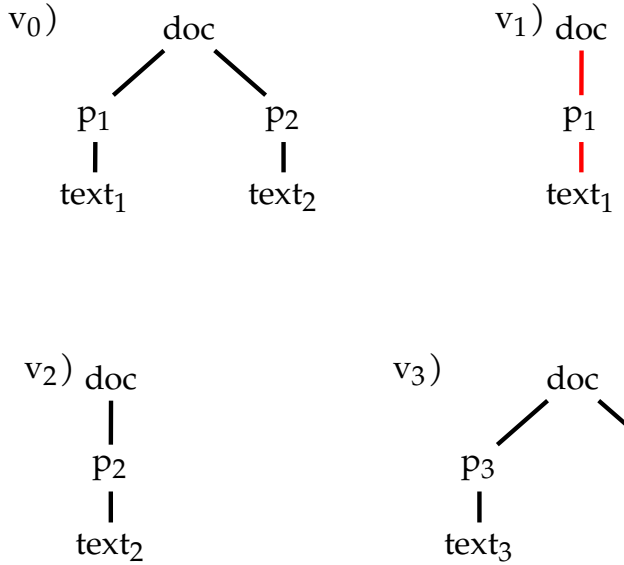
$Pr(e_2)=0.7$; $Pr(e_3)=0.5$

Result of the merge ($\mathcal{P}$)

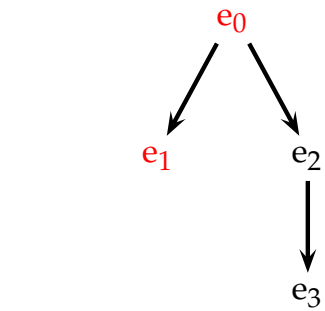
VERSIONING WITH THE P-DOCUMENT $\mathcal{P}$

- In the initial state, content of $\mathcal{P}$ is that of v_0 with $f_{p_1} = e_0$ and $f_{p_2} = e_0$

Example of the versioning process

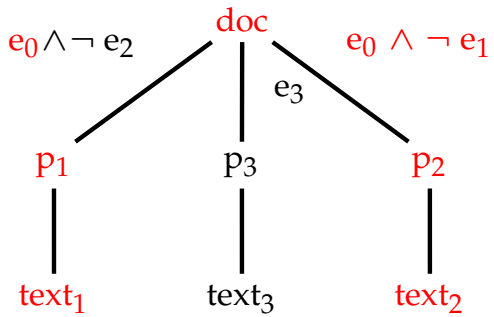


Versions with Uncertain Data



$\mathcal{L} = \{(v_0, v_1), (v_0, v_2), (v_2, v_3)\}$

Graph of version derivations



$Pr(e_0)=0.5$; $Pr(e_1)=0.4$

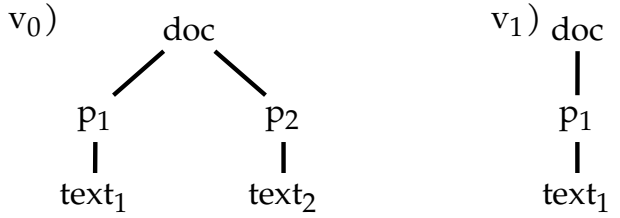
$Pr(e_2)=0.7$; $Pr(e_3)=0.5$

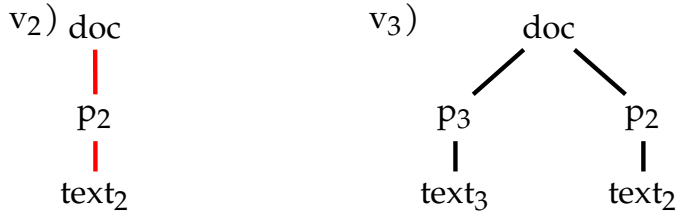
Result of the merge ($\mathcal{P}$)

VERSIONING WITH THE P-DOCUMENT $\mathcal{P}$

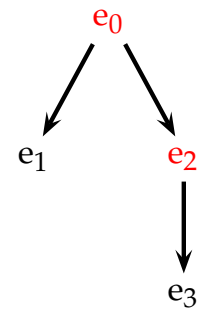
- In the initial state, content of $\mathcal{P}$ is that of v_0 with $f_{p_1} = e_0$ and $f_{p_2} = e_0$
- In the second state, content of $\mathcal{P}$ does not change, but $f_{p_2} = e_0 \wedge \neg e_1$ because subtree rooted at p_2 has removed in v_1

Example of the versioning process



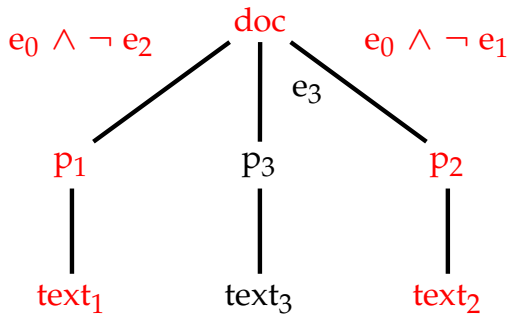


Versions with Uncertain Data



$\mathcal{L} = \{(v_0, v_1), (v_0, v_2), (v_2, v_3)\}$

Graph of version derivations



$Pr(e_0)=0.5$; $Pr(e_1)=0.4$

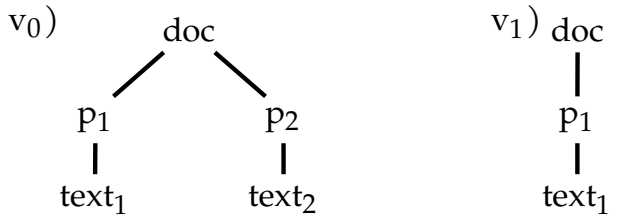
$Pr(e_2)=0.7$; $Pr(e_3)=0.5$

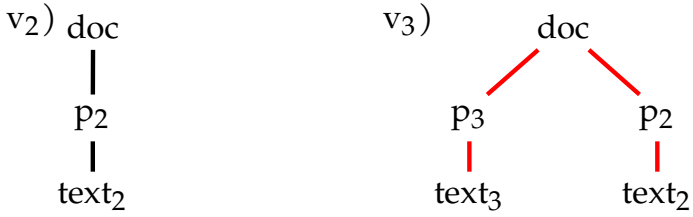
Result of the merge ($\mathcal{P}$)

VERSIONING WITH THE P-DOCUMENT $\mathcal{P}$

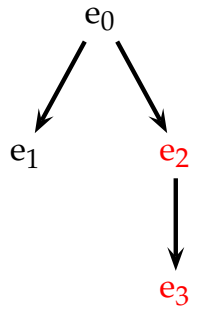
- In the initial state, content of $\mathcal{P}$ is that of v_0 with $f_{p_1} = e_0$ and $f_{p_2} = e_0$
- In the second state, content of $\mathcal{P}$ does not change, but $f_{p_2} = e_0 \wedge \neg e_1$ because subtree rooted at p_2 has removed in v_1
- In the third state, content of $\mathcal{P}$ remains unchanged, but $f_{p_1} = e_0 \wedge \neg e_2$ because subtree rooted at p_1 has removed in v_2

Example of the versioning process



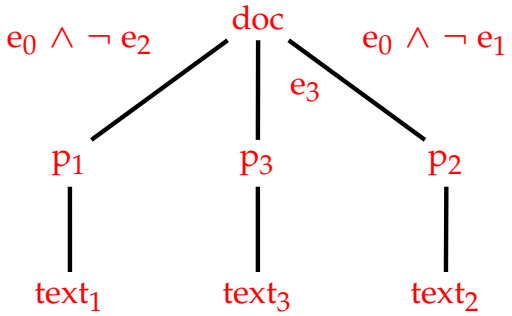


Versions with Uncertain Data



$\mathcal{L} = \{(v_0, v_1), (v_0, v_2), (v_2, v_3)\}$

Graph of version derivations



$Pr(e_0)=0.5$; $Pr(e_1)=0.4$

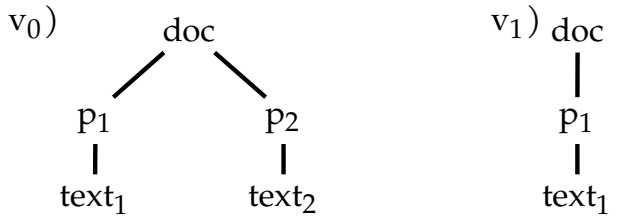
$Pr(e_2)=0.7$; $Pr(e_3)=0.5$

Result of the merge ($\mathcal{P}$)

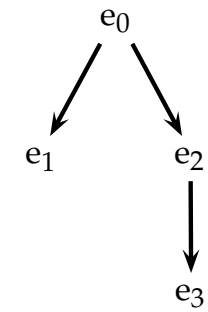
VERSIONING WITH THE P-DOCUMENT $\mathcal{P}$

- In the initial state, content of $\mathcal{P}$ is that of v_0 with $f_{p_1} = e_0$ and $f_{p_2} = e_0$
- In the second state, content of $\mathcal{P}$ does not change, but $f_{p_2} = e_0 \wedge \neg e_1$ because subtree rooted at p_2 has removed in v_1
- In the third state, content of $\mathcal{P}$ remains unchanged, but $f_{p_1} = e_0 \wedge \neg e_2$ because subtree rooted at p_1 has removed in v_2
- In the last state, $\mathcal{P}$ is updated with new subtree of v_3 rooted at p_3 and $f_{p_3} = e_3$

Example of the versioning process

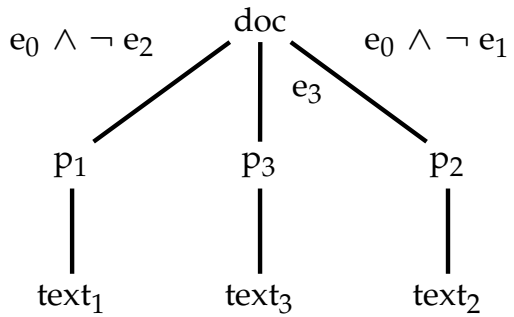


Versions with Uncertain Data



$\mathcal{L} = \{(v_0, v_1), (v_0, v_2), (v_2, v_3)\}$

Graph of version derivations



$Pr(e_0)=0.5$; $Pr(e_1)=0.4$

$Pr(e_2)=0.7$; $Pr(e_3)=0.5$

Result of the merge ($\mathcal{P}$)

Semantics of events

- Variables can represent reliability of contributors.
- Probabilities can be seen as reliability values.

Example of Querying the system

$\mathcal{Q}$: selecting of version v_0

$\mathcal{Q}'$: choose valuation ξ setting e_0 to true and others to false

In the model, $\mathcal{Q} \Leftrightarrow \mathcal{Q}'$ and evaluation is done on $\mathcal{P}$

Outline

Open problems

- An appropriate XML diff method to explore
- Extension of basic operations on classical models to our intended one
- An extensive experimentation on real data e.g., data from Wikipedia
- Study reliability of detecting controversial topics in online collaborative environments

- XCC[RB10] is efficient for collaborative contexts; it seems to be a promising basis
- A possible approach to implement basic operations is to use probabilistic events
- Performance evaluations of our setting, e.g., running time of the prob merging algorithm

MERCI !


Talel Abdessalem, M. Lamine Ba, and Pierre Senellart.
A probabilistic XML merging tool.
In *Proc. EDBT*, 2011.


Serge Abiteboul, Benny Kimelfeld, Yehoshua Sagiv, and Pierre Senellart.
On the expressiveness of probabilistic XML models.
VLDB Journal, 18:1041–1064, 2009.


Ander de Keijzer and Maurice van Keulen.
IMPrECISE: Good-is-good-enough data integration.
In *Proc. ICDE*, 2008.


Evgeny Kharlamov, Werner Nutt, and Pierre Senellart.
Updating probabilistic XML.
In *Proc. Updates in XML*, 2010.


Tancred Lindholm, Jaakko Kangasharju, and Sasu Tarkoma.
Fast and simple XML tree differencing by sequence alignment.
In *Proc. DocEng*, 2006.


Amelie Marian.
Detecting changes in XML documents.
In *Proc. ICDE*, 2002.


Silviu Maniu, Bogdan Cautis, and Talel Abdessalem.
Building a signed network from interactions in Wikipedia.
In *Databases and Social Networks*, 2011.


Sebastian Rönnau and Uwe Borghoff.
XCC: change control of XML documents.
Computer Science - Research and Development, pages 1–17, 2010.


Maurice van Keulen, Ander de Keijzer, and Wouter Alink.
A probabilistic XML approach to data integration.
In *Proc. ICDE*, 2005.