Presentation of Update Semantics of Relational Views

Nhung Ngo & Yu Liu

FCCOD 2014
Outline

1 Problem

- Overview of the problem being addressed
- Formal definition of the problem

2 Solution

- Translation under constant complement
- Update policy
- Advantages and disadvantages of solutions
Overview
Overview
Overview

Problem Overview of the problem being addressed

Nhung Ngo & Yu Liu (FCCOD 2014) Presentation of Update Semantics of Relational Views

[Diagram showing view definitions flowing from database update to view update]
Overview of the problem being addressed:

- Database update
- View definitions
- View update
- View definitions
Example 1
Unexpected changes on the view

\[ V = E \bowtie D \]

<table>
<thead>
<tr>
<th>EMP</th>
<th>DEP</th>
<th>MGR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mike</td>
<td>EEE</td>
<td>Susan</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EMP</th>
<th>DEP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mike</td>
<td>EEE</td>
</tr>
<tr>
<td>Mary</td>
<td>CS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DEP</th>
<th>MGR</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEE</td>
<td>Susan</td>
</tr>
</tbody>
</table>
Example 1
Unexpected changes on the view

\[ V = E \bowtie D \]

<table>
<thead>
<tr>
<th>EMP</th>
<th>DEP</th>
<th>MGR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mike</td>
<td>EEE</td>
<td>Susan</td>
</tr>
<tr>
<td>Jane</td>
<td>CS</td>
<td>Alex</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EMP</th>
<th>DEP</th>
<th>MGR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mike</td>
<td>EEE</td>
<td>Susan</td>
</tr>
<tr>
<td>Mary</td>
<td>CS</td>
<td></td>
</tr>
</tbody>
</table>
### Example 1

Unexpected changes on the view

\[ V = E \bowtie D \]

<table>
<thead>
<tr>
<th>EMP</th>
<th>DEP</th>
<th>MGR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mike</td>
<td>EEE</td>
<td>Susan</td>
</tr>
<tr>
<td>Jane</td>
<td>CS</td>
<td>Alex</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EMP</th>
<th>DEP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mike</td>
<td>EEE</td>
</tr>
<tr>
<td>Mary</td>
<td>CS</td>
</tr>
<tr>
<td>Jane</td>
<td>CS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DEP</th>
<th>MGR</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEE</td>
<td>Susan</td>
</tr>
<tr>
<td>CS</td>
<td>Alex</td>
</tr>
</tbody>
</table>
Example 1
Unexpected changes on the view

\[ V = E \bowtie D \]

<table>
<thead>
<tr>
<th>EMP</th>
<th>DEP</th>
<th>MGR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mike</td>
<td>EEE</td>
<td>Susan</td>
</tr>
<tr>
<td>Jane</td>
<td>CS</td>
<td>Alex</td>
</tr>
<tr>
<td>Mary</td>
<td>CS</td>
<td>Alex</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EMP</th>
<th>DEP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mike</td>
<td>EEE</td>
</tr>
<tr>
<td>Mary</td>
<td>CS</td>
</tr>
<tr>
<td>Jane</td>
<td>CS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DEP</th>
<th>MGR</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEE</td>
<td>Susan</td>
</tr>
<tr>
<td>CS</td>
<td>Alex</td>
</tr>
</tbody>
</table>
Example 1
Unexpected changes on the view

\[ V = E \Join D \]

<table>
<thead>
<tr>
<th>EMP</th>
<th>DEP</th>
<th>MGR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mike</td>
<td>EEE</td>
<td>Susan</td>
</tr>
<tr>
<td>Jane</td>
<td>CS</td>
<td>Alex</td>
</tr>
<tr>
<td>Mary</td>
<td>CS</td>
<td>Alex</td>
</tr>
</tbody>
</table>

The changes on the db must reflect \textit{exactly} the changes on the view.
### Example 2
Unjustified changes on the database

\[ V = \pi_{\text{EMP}, \text{DEP}}(R) \]

<table>
<thead>
<tr>
<th>EMP</th>
<th>DEP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mike</td>
<td>EEE</td>
</tr>
<tr>
<td>Jane</td>
<td>CS</td>
</tr>
</tbody>
</table>

\[ R \]

<table>
<thead>
<tr>
<th>EMP</th>
<th>DEP</th>
<th>MGR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mike</td>
<td>EEE</td>
<td>Susan</td>
</tr>
<tr>
<td>Jane</td>
<td>CS</td>
<td>Alex</td>
</tr>
<tr>
<td>Jane</td>
<td>CS</td>
<td>Max</td>
</tr>
</tbody>
</table>
Example 2
Unjustified changes on the database

\[ V = \pi_{\text{EMP}, \text{DEP}}(R) \]

<table>
<thead>
<tr>
<th>EMP</th>
<th>DEP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mike</td>
<td>EEE</td>
</tr>
<tr>
<td>Jane</td>
<td>CS</td>
</tr>
</tbody>
</table>

\[ (\text{Jane, CS}) \]

\[ R \]

<table>
<thead>
<tr>
<th>EMP</th>
<th>DEP</th>
<th>MGR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mike</td>
<td>EEE</td>
<td>Susan</td>
</tr>
<tr>
<td>Jane</td>
<td>CS</td>
<td>Alex</td>
</tr>
<tr>
<td>Jane</td>
<td>CS</td>
<td>Max</td>
</tr>
</tbody>
</table>
Example 2
Unjustified changes on the database

\[ V = \pi_{EMP, DEP}(R) \]

<table>
<thead>
<tr>
<th>EMP</th>
<th>DEP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mike</td>
<td>EEE</td>
</tr>
<tr>
<td>Jane</td>
<td>CS</td>
</tr>
</tbody>
</table>

\( (Jane, CS) \)

R

<table>
<thead>
<tr>
<th>EMP</th>
<th>DEP</th>
<th>MGR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mike</td>
<td>EEE</td>
<td>Susan</td>
</tr>
<tr>
<td>Jane</td>
<td>CS</td>
<td>Alex</td>
</tr>
<tr>
<td>Jane</td>
<td>CS</td>
<td>Max</td>
</tr>
</tbody>
</table>
Example 2
Unjustified changes on the database

\[ V = \pi_{\text{EMP}, \text{DEP}}(R) \]

<table>
<thead>
<tr>
<th>EMP</th>
<th>DEP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mike</td>
<td>EEE</td>
</tr>
<tr>
<td>Jane</td>
<td>CS</td>
</tr>
</tbody>
</table>

\( (\text{Jane}, \text{CS}) \)

\[ R \]

<table>
<thead>
<tr>
<th>EMP</th>
<th>DEP</th>
<th>MGR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mike</td>
<td>EEE</td>
<td>Susan</td>
</tr>
<tr>
<td>Jane</td>
<td>CS</td>
<td>Alex</td>
</tr>
<tr>
<td>Jane</td>
<td>CS</td>
<td>Max</td>
</tr>
</tbody>
</table>

Modify the database \textit{only if required} to reflect the changes on the view.
Basic Notation

\( S \) database schema - set of all database instances (\textit{database states})

\( T \) set of all view instances (\textit{view states})

- view \( f : S \rightarrow T \)
- view update \( u : T \rightarrow T \)
- database update \( d : S \rightarrow S \)

\( U_1 \) set of all database updates

\( U_f \) set of all view updates
Concepts to be formalized

- Q1: Given a view update $u$, what are the constraints on the database update that translates $u$?
Q1: Given a view update $u$, what are the constraints on the database update that translates $u$?

Q2: What sets of view updates do we want to translate, that is, what sets of updates users are to be allowed on the view?
Concepts to be formalized

- Q1: Given a view update $u$, what are the constraints on the database update that translates $u$?
- Q2: What sets of view updates do we want to translate, that is, what sets of updates users are to be allowed on the view?
- Q3: How do we associate with each view update a database update that translates it?
A database update $d$ is a *translation* of a view update $u$ iff for each database state $s \in S$

1. $uf(s) = fd(s)$ \hspace{2cm} (consistent)
2. $uf(s) = f(s) \rightarrow d(s) = s$ \hspace{2cm} (acceptable)
Definitions

Q1: A translation of a view update

A database update $d$ is a translation of a view update $u$ iff for each database state $s \in S$

1. $uf(s) = fd(s)$  
   (consistent)
2. $uf(s) = f(s) \rightarrow d(s) = s$  
   (acceptable)
Definitions

Q1: A translation of a view update

A database update $d$ is a *translation* of a view update $u$ iff for each database state $s \in S$

1. $uf(s) = fd(s)$  \hspace{5cm} \text{(consistent)}
2. $uf(s) = f(s) \rightarrow d(s) = s$  \hspace{5cm} \text{(acceptable)}
A set $U$ of view updates is called *complete* iff
A set $U$ of view updates is called *complete* iff

(1) $\forall u, v \in U, uv \in U$
A set $U$ of view updates is called complete iff

1. $\forall u, v \in U, uv \in U$
2. $\forall s \in S, \forall u \in U, \exists u' \in U u'u_f(s) = f(s)$
Definitions
Q3: A translator

A mapping $T : U \rightarrow U_1$ is called a \textit{translator} iff

(1) $\forall u \in U$, $T_u$ is a translation of $u$

(2) $\forall u, v \in U$, $T_{uv} = T_u T_v$

The view update problem

Given a complete set $U$ of view updates, find a translator of $U$
Intuitive idea

- If the view $f$ is injective

\[ f(s) \]

- If the view $f$ is not injective

Need a view complement $g$ of $f$ so that $f \times g$ is injective
If the view $f$ is injective

If the view $f$ is not injective

Need a view complement $g$ of $f$ so that $f \times g$ is injective
Intuitive idea

- If the view $f$ is injective
  
  ![Diagram of injective function]

  
  If the view $f$ is not injective
  
  Need a view complement $g$ of $f$ so that $f \times g$ is injective
The view complement

\[ g \text{ is a complement of } f \text{ iff } f \times g = 1 \]

\[ g \text{ is a complement of } f \text{ iff } \forall s, s' \in S_\Sigma, s \neq s' \wedge f(s) = f(s') \rightarrow g(s) \neq g(s') \]

- A complement of \( f \) contains “the information not visible within \( f \)”
- A complement of \( f \) is able to distinguish database states that \( f \) maps to the same view state
- A view complement always exists (a renamed copy of the whole db schema in the worst case)
- In general, there is no unique minimal complement
The view complement

An example
The view complement
An example
The view complement

An example

\[ g \quad s_1 \quad f \quad t_1 \]

\[ g \quad s_2 \quad f \quad t_2 \]

\[ g \quad s_3 \quad f \quad t_2 \]
Rectangle Rule from Chamberlin et al (1975): ”An insertion, deletion, or update via a view must affect only information visible within the rectangle of the view.”

- A complement $g$ of a view update $u$ should not be changed (i.e. invariant) by a database update.
- A translation $\gamma_u$ of $u$ should be verified that it makes $g$ invariant.
A given database, a view and a complement view

<table>
<thead>
<tr>
<th>E(EMP, DEP)</th>
<th>M(DEP, MGR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1: EMP → DEP</td>
<td>C2: DEP ↔ MGR</td>
</tr>
<tr>
<td>C3: E[DEP] = M[DEP]</td>
<td></td>
</tr>
</tbody>
</table>

\[
f(s) = \pi_{\text{DEP},\text{MGR}} E \bowtie M
\]

\[
g(s) = M
\]

<table>
<thead>
<tr>
<th>s =</th>
<th>E</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMP</td>
<td>DEP</td>
<td>DEP</td>
</tr>
<tr>
<td>Mary</td>
<td>CS</td>
<td>CS</td>
</tr>
<tr>
<td>Jane</td>
<td>CS</td>
<td>EEE</td>
</tr>
<tr>
<td>Mike</td>
<td>EEE</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>f =</th>
<th>EM</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEP</td>
<td>MGR</td>
</tr>
<tr>
<td>Mary</td>
<td>Alex</td>
</tr>
<tr>
<td>Jane</td>
<td>Alex</td>
</tr>
<tr>
<td>Mike</td>
<td>Susan</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>g =</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEP</td>
<td>MGR</td>
</tr>
<tr>
<td>CS</td>
<td>Alex</td>
</tr>
<tr>
<td>EEE</td>
<td>Susan</td>
</tr>
</tbody>
</table>
Example of a translation that leaves a complement invariant

\(u\): Replace employee Mary by employee John.

\(g\): Table M.

\(\gamma_u\): 
\[E = (M^*u(EM))[EMP,DEP];M = M.\]

\[\begin{array}{c|c}
\text{EMP} & \text{DEP} \\
\hline
\text{Mary} & \text{CS} \\
\text{Jane} & \text{CS} \\
\text{Mike} & \text{EEE} \\
\end{array}\]

\[\begin{array}{c|c|c}
\text{DEP} & \text{MGR} & \text{EMP} \\
\hline
\text{CS} & \text{Alex} & \text{Mary} \\
\text{EEE} & \text{Susan} & \text{Jane} \\
\end{array}\]

\(\gamma_u\) \(\rightarrow\) 

\[\begin{array}{c|c|c|c}
\text{EMP} & \text{DEP} & \text{MGR} \\
\hline
\text{John} & \text{CS} & \text{Alex} \\
\text{Jane} & \text{CS} & \text{Mary} \\
\text{Mike} & \text{EEE} & \text{Susan} \\
\end{array}\]

\(f\) = 

\[\begin{array}{c|c|c|c|c}
\text{DEP} & \text{MGR} & \text{EMP} & \text{EM} \\
\hline
\text{Mary} & & \text{Alex} \\
\text{Jane} & & \text{Alex} \\
\text{Mike} & & \text{Susan} \\
\end{array}\]

\(u\) \(\rightarrow\) 

\[\begin{array}{c|c|c|c|c}
\text{DEP} & \text{MGR} & \text{EMP} & \text{EM} \\
\hline
\text{John} & & \text{Alex} \\
\text{Jane} & & \text{Alex} \\
\text{Mike} & & \text{Susan} \\
\end{array}\]
$u$ is $g$ – translatable

$u$ is $g$ – translatable iff for all $s$ in $S$, there exists a $s'$ that:

1. $f(s') = uf(s)$
2. $g(s') = g(s)$

The composition of $g$ – translatable updates is also $g$ – translatable

- $u, v$ are $g$ – translatable $\Rightarrow uv$ is $g$ – translatable
  
  1. $f(s'') = u(f(s')) = uvf(s)$
  2. $g(s'') = g(s') = g(s)$
Example of a composition of $g$ — *translatable* updates

$u$: Replace employee Mary by employee John.

$v$: Replace employee Jane by employee Lewis.

$g$: Table M.

$\gamma_{uv}: E = (M^{*uv}(EM))[EMP,DEP]; M = M.$
$g$-translation: $\gamma_u$

For a given $f, g, u$ if $u$ is $g$-translatable then $\gamma_u = (f \times g)^{-1}(uf \times g)$.

$\gamma_u$ is a translation of $u$ ($\gamma_u$ is called a $g$-translation of $u$):
- $uf = f\gamma_u \rightsquigarrow$ Consistent
- $\gamma_u(s) = s \rightsquigarrow$ Acceptable

$\gamma_u$ leaves $g$ invariant
- $g\gamma_u = g$

If $u$ is $g$-translatable, $\gamma_u$ always exists and is unique.
How to choose a complement of a view?(1)

The choice of $g$ impacts that whether $u$ is $g$ – translatable or not.

$u$: Replace employee Mary by employee John.
$g'$: Table E.
$\gamma_u : E = (M^u(EM))[\text{EMP,DEP}]; M = M$.

\[
\begin{array}{ccc}
E & M & E \\
\hline
\text{EMP} & \text{DEP} & \text{DEP} & \text{MGR} \\
\text{Mary} & \text{CS} & \text{CS} & \text{Alex} \\
\text{Jane} & \text{CS} & \text{EEE} & \text{Susan} \\
\text{Mike} & \text{EEE} & & \\
\end{array}
\]

\[
\begin{array}{ccc}
\gamma_u & s' & \\
\hline
\rightarrow & \text{EMP} & \text{DEP} \\
\text{John} & \text{CS} & \text{Alex} \\
\text{Jane} & \text{CS} & \text{Susan} \\
\text{Mike} & \text{EEE} & \\
\end{array}
\]

\[
\begin{array}{ccc}
EM & M & EM \\
\hline
\text{DEP} & \text{MGR} & \text{DEP} & \text{MGR} \\
\text{Mary} & \text{Alex} & \text{John} & \text{Alex} \\
\text{Jane} & \text{Alex} & \text{Jane} & \text{Alex} \\
\text{Mike} & \text{Susan} & \text{Mike} & \text{Susan} \\
\end{array}
\]

\[
f = \frac{EM}{\text{DEP} \text{MGR}} \\
uf = \frac{EM}{\text{DEP} \text{MGR}}
\]
How to choose a complement of a view?(2)

The choice of $g$ impacts that whether $u$ is $g$-translatable or not.

$w$: Permute the managers.

$g'$: Table E.

$\gamma_w: E = E; M = (E^*w(EM))[EMP,DEP]$.

$$
\begin{array}{c|c|c}
\text{EMP} & \text{DEP} & \gamma_w \rightarrow \text{EMP} \\
\text{Mary} & \text{CS} & \text{Mary} \\
\text{Jane} & \text{CS} & \text{Jane} \\
\text{Mike} & \text{EEE} & \text{Mike} \\
\end{array}
\quad
\begin{array}{c|c|c}
\text{DEP} & \text{MGR} & \text{DEP} \\
\text{CS} & \text{Alex} & \text{CS} \\
\text{EEE} & \text{Susan} & \text{EEE} \\
\end{array}
\quad
\begin{array}{c|c|c|c|c}
\text{EMP} & \text{DEP} & \text{MGR} & \text{EMP} & \text{DEP} \\
\text{Mary} & \text{CS} & \text{Susan} & \text{Mary} & \text{CS} \\
\text{Jane} & \text{CS} & \text{Susan} & \text{Jane} & \text{CS} \\
\text{Mike} & \text{EEE} & \text{Alex} & \text{Mike} & \text{EEE} \\
\end{array}
\quad
\begin{array}{c|c|c}
\text{DEP} & \text{MGR} & \text{DEP} \\
\text{CS} & \text{Susan} & \text{CS} \\
\text{EEE} & \text{Alex} & \text{EEE} \\
\end{array}
$$
How to choose a complement of a view?(3)

For a given view $u$, $g$, $h$ are both complements of $f$ and $h$ contains less information than $g$. If $u$ is $g$–translatable then:

1. $u$ is also $h$–translatable
2. $h$–translation $= g$–translation

The set of $g$–translatable updates is maximal when the complement $g$ is minimal.

- We could like to find the minimal complements, so that get maximal update sets (a minimal complement is not unique).

A complement view: an update policy
Universal property of translation under constant complement

Given a complete set $U \subset U_f$, a view $f$ and a complement view $g$ of $f$:

This paper provided translators $T$ for $U$ if $\forall u \in U$: $u$ is $g$–translatable:

1. Select a complement $g$ of the given view $f$.
2. Verify that view updates of the given complete set $U$ make $g$ invariant.
3. For each view update $u \in U$, the translation $T_u = (f \times g)^{-1}(uf \times g)$.

For every $T$ of $U$, there exists a complement $g$ that:

1. $\forall u \in U$, $u$ is $g$–translatable.
2. $\forall u \in U$, $g$–translation $\gamma_u = (f \times g)^{-1}(uf \times g)$
Advantages & Disadvantages

Advantages:

1. This paper provides a formal framework for solving the view update problem.
2. The method is beneficial for solving view update issues in Data Integration.

Disadvantages:

1. Too theoretical, no algorithms for implementation from practical point of view.
2. This paper does not show how to find a minimal complement.
Related works

- Lechtenboerger (2003) gives a characterisation of the constant complement principle in terms of “undo” operations in SQL server.
- Cosmadakis and Papadimitriou (1984) consider a restricted setting that consists of a single database relation and two views defined by projections.
- Gottlob et al (1988) extend to the class of so-called consistent views, which properly contains the views translating under constant complement. The complement is not required to remain invariant in their framework.
- Enrico Franconi and Paolo Guagliardo [2011] provide a general framework for view updating (under constraints) based on the notion of determinacy.