Commutative Replicated Data Types for the Semantic Web

Pascal Molli
Kickoff Kolflow
Feb 2011

What is CRDTs

- CRDTs stands for “Commutative Replicated Data Types”
- It is a class of replication algorithms that belongs to the optimistic replication approach
- Optimistic replication has many applications:
  - All synchronization problems, including cache…
  - High availability of data in NoSQL stores for the Cloud…
  - Version Control System (centralized or decentralized…)
  - Real-time editing system…

CRDT and Kolflow ?

- Kolflow aims to build a Social Semantic Space where humans and machines will collaboratively edit contents and knowledge...
- This generates huge synchronization problem for text, for semantic datasets and for both...
  - Consistency ?
  - Scalability ?
  - How to manage concurrent modifications issued by humans and machines ?

CRDT in action : DSMW

- DSMW means Distributed Semantic MediaWiki. DSMW is based on Logoot CRDT.
- It allows to compose a network of semantic mediawiki server and manage synchronisation of data and text between servers.
- Data and text can be modified concurrently at any point of the network,
- Synchronization algorithms ensure that if all servers received the same operations, then all servers are in the same state.

http://doi.ieeecomputersociety.org/10.1109/TPDS.2009.173

How it works ? 2 pages with [[is in::tutorial]] annotations

page | discussion | edit | history | article admin (1 patches)

Hello

A "Hello World" program is a computer program which prints out "Hello, world!" on a display device. It is used in many introductory tutorials for teaching a prog
simplest programs possible in a computer language. Some are surprisingly complex, especially in some graphical user interface (GUI) contexts, but most are
command line
a LED being tu

page | discussion | edit | history | article admin (2 patches)

World

Ein Hallo-Welt-Programm ist ein kleines Computerprogramm und soll auf möglichst einfache Weise zeigen, welche Anweisungen oder Besta
werden und somit einen ersten Einblick in die Syntax geben. Aufgabe des Programms ist, den Text Hallo Welt! oder auf Englisch Hello, worl
Installation eines Compilers für die entsprechende Programmiersprache zu überprüfen. Aufgrund der einfachen Aufgabenstellung kann ein
zumeist nur dem Programmierparadigma der imperativen Programmierung und demonstriert somit zwar nur einen Bruchteil der Möglichke

Die Verwendung des Textes „Hello, world!", der auch durch einen beliebigen Text ersetzt werden kann, aber dennoch gerne unverändert ber
Bell Laboratories über die Programmiersprache C zurück, das Brian Kernighan 1974 verfasste, nachdem er dort schon ein Jahr zuvor die Wo
hatte[1]. Bekanntheit erlangte der Text jedoch erst durch die Veröffentlichung in dem Buch The C Programming Language (deutsch: Progra
Beispiel die Schreibung „hello, world" verwendet wurde. tutorial

[[isin::tutorial]]

Create your feeds

page | discussion | edit | history | article admin (0 patches)

Administration push site addition

Add a new pushfeed:

PushFeed Name:

PushTutorial

Request:

```
[[is in::tutorial]]
```

ADD

special page

P2P Administration

PULL: [Add]

PUSH: [Add] [Remove] [Push]

Site	Pages	All patches	Published Patches	Unpublished Patches
PushFeed:PushTutorial	[2]	[3]	[0]	[3]

Patches are semantics pages

page | discussion | edit | history | article admin (1 patches)

DSMW on Hello

List of patchs

Wed, 19 Aug 2009 08:08:27 GMT : 2 insert, 0 delete (Patch:BC0A6E356ABF4395C83A525B40645F0A13)

List of pushs

PushFeed:PushTutorial : all Hello'patchs are pushed

[Push page : "Hello"]

page | discussion | edit | history | article admin (2 patches)

DSMW on World

List of patchs

Wed, 19 Aug 2009 08:11:20 GMT : 4 insert, 0 delete (Patch:BC0A6E356ABF4395C83A525B40645F0A16)

Wed, 19 Aug 2009 08:11:20 GMT : 1 insert, 1 delete (Patch:BC0A6E356ABF4395C83A525B40645F0A21)

List of pushs

PushFeed:PushTutorial : all World'patchs are pushed

[Push page : "World"]

Now subscription and pull from another DSMW node...

page | discussion | edit | history | article admin (0 patches)

Administration pull site addition

Add a new site:

Server Url:

http://localhost/wiki1

PushFeed Name:

PushTutorial

PullFeed Name:

PullTutorial

ADD

special page

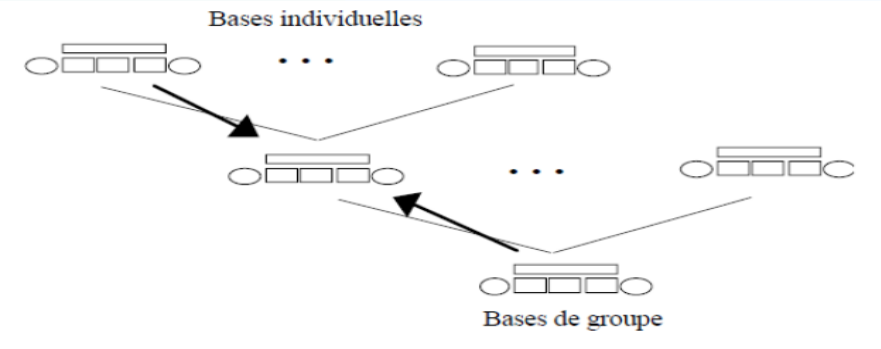
P2P Administration

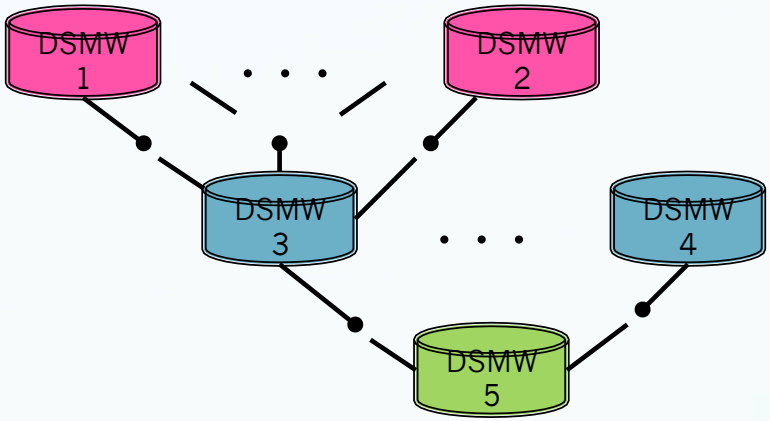
PULL: [Add][Remove] [Pull]

Site	Pages	Remote Patches	Local Patches
PullFeed:PullTutorial	[0]	[3]	[0]

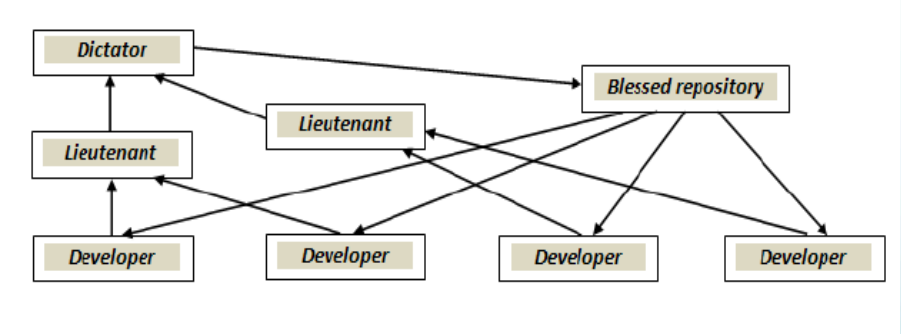
PUSH: [Add]

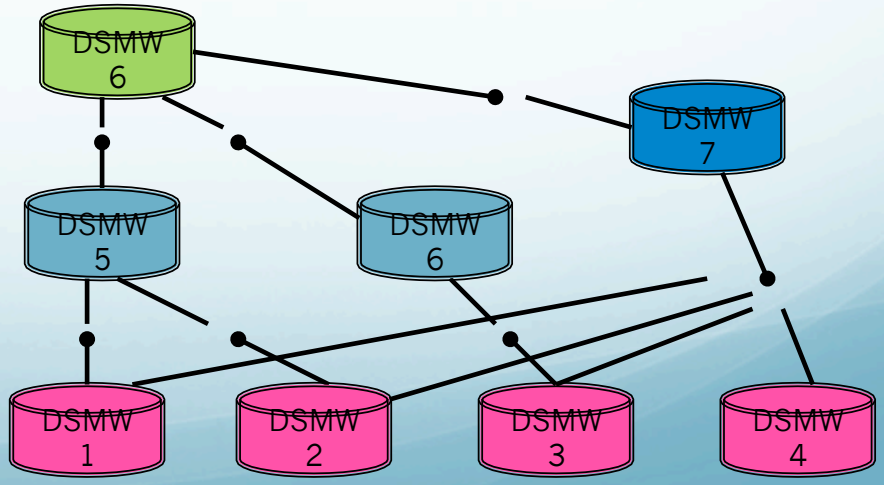
DSMW: processes support





CO4 ontology engineering





Dictator-Lieutenant process in software engineering

DSMW highlights

- DSMW is the first tool of social semantic web based on a CRDT.
  - Improves scalability and allows new collaboration models
- Demonstrates how it is possible to embed a CRDT algorithm in a real working system (Semantic media wiki) with no modification of the original system.
- Wise2009, ESWC2010(demo),DocEng2010 (demo)
- Available on http://dsmw.org

DSMW and optimistic replication

- DSMW instantiate the optimistic replication model
- Suppose a unknown number of semantic wiki servers:
  - An operation is performed at one site :
    - **Ex:** add a new line that contain a semantic annotation. Operation is executed immediately
  - Operation is propagated to other sites
    - in DSMW using publish/subscribe model, with the notion feeds of operations...
  - Operation is received by other sites and integrated with concurrent modification.
- The system (all the network) is correct if it ensures Causality, Convergence and Intentions...

Modèle de cohérence CCI [SJZ +98]

- Causalité :
  - Les opérations ordonnées par une relation de précédence de Lamport sont exécutées dans le même ordre sur tous les sites.
- Convergence :
  - Lorsque le même ensemble d'opérations a été exécuté sur tous les sites, toutes les copies du document partagé sont identiques.
- Intention:

  Pour toute opération op, les effets de l'exécution de op sur tous les sites sont les mêmes que les effets de l'exécution de l'opération op sur son état de génération.

CCI Formalisation...

- Préservation de la causalité :

$$\forall S = (H_S, \rightarrow_S, R_S) \in \mathbb{S}, ((H_S \times H_S) \cap \rightarrow) \subseteq \rightarrow_S$$

- Convergence :

$$\forall S = (H_S, \rightarrow_S, n) \in \mathbb{S}. \forall S' = (H_{S'}, \rightarrow_{S'}, n') \in \mathbb{S}.$$
$$(H_S = H_{S'} \Rightarrow D(S) = D(S'))$$

- Préservation de l'intention (texte):

$$\forall S \in \mathbb{S}, (E, \prec) = I(S) \wedge (E_S, \prec_S) = D(S) \Rightarrow (E = E_S) \wedge (\prec \subseteq \prec_S)$$

CRDTs

- If the execution of concurrent operations is commutative, then whatever the order of reception of operation on sites, sites eventually converge :
  - D(S) o O1 o O2 = D(S) o O2 o O1
- Pb : Data types have operations that does not commute:
  - a set {} with Insert(elt) and delete(elt) is cheap to implement but does not commute…
  - A sequence with insert(x,1)/delete(1) does not commute
- Logoot is CRDT for sequence…

Logoot

- Associer à chaque élément un identifiant:
  - $<p_1, s_1, h_1><p_2, s_2, h_2>...<p_k, s_k, h_k>$
    - p_i est un entier ($0 < p_i < BASE$)
    - s_i est un identifiant de réplique
    - h_i est la valeur de l'horloge logique
- Ordonnés suivant un ordre lexicographique,
- Unique dans le temps et l'espace,
- Ensemble des identifiants est dense.

Document Logoot (1)

Réplique 1

Table des identifiants	Document
<0, NA, NA>	
<15, 1, 1>	A
<29, NA, NA>	

Insert(<15, 1, 1>, A)

Table des identifiants	Document
<0, NA, NA>	
<15, 1, 1>	A
<29, NA, NA>	

Réplique 2

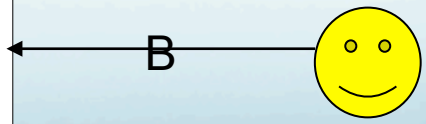
Document Logoot (2)

Réplique 1

Table des identifiants	Document
<0, NA, NA>	
<15, 1, 1>	A
<15, 1, 1><15, 1, 2	X
<15, 2, 1>	B
<29, NA, NA>	

Insert(<15, 1, 1><15, 1, 2>, X)

Table des identifiants	Document
<0, NA, NA>	
<15, 1, 1>	A
<15, 1, 1><15, 1, 2	X
<15, 2, 1>	B
<29, NA, NA>	



Réplique 2

Complexité moyenne

	Génération	Intégration		Complexité en espace
	Insertion/Suppression	Insertion	Suppression	
Logoot	$O(k)$	$O(k.log(n))$	$O(k.log(n))$	$O(k.n)$
WOOT	$O(N)$	$O(N^2)$	$O(N)$	$O(N)$
MOT2	$O(N)$	$O(N)$	$O(N)$	$O(N)$
TreeDoc	$O(K.log(N))$	$O(K.log(N))$	$O(K.log(N))$	$O(N)$

- k : longueur d'un identifiant
- n : nombre de lignes
- N : nombre d'opérations
- K : taille d'un supernœud
- L'approche est valable uniquement si k est petit.

Logoot et Wikipedia

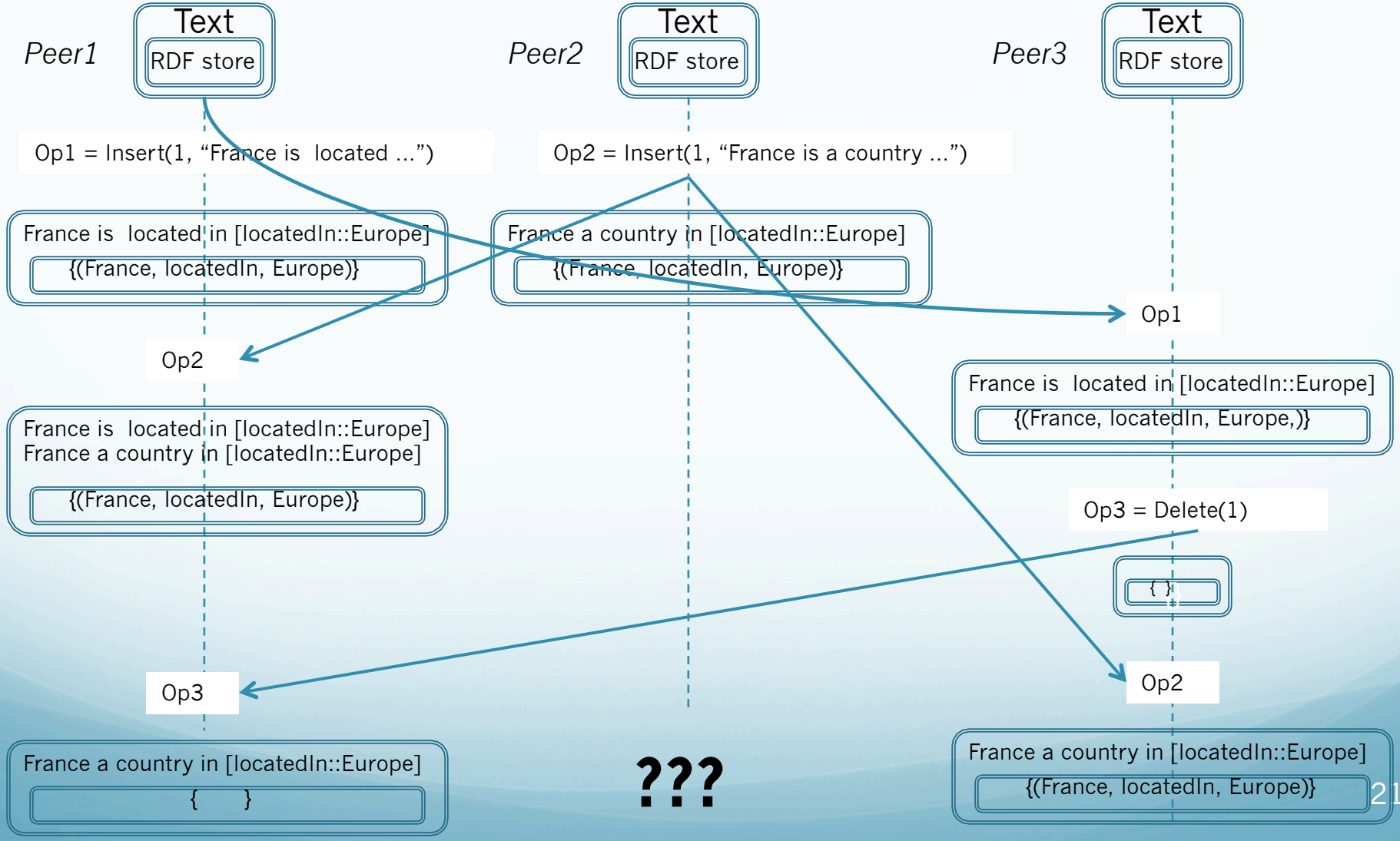
Logoot	Pages les plus éditées	Articles les plus édités	Articles les plus longs	Articles de qualité	Articles choisi aléatoirement
Stratégie aléatoire	12,2	1,0	1,3	1,0	1,0

- Pages « normales » : k valeur minimale
- Pages extrêmes :
  - Articles : $k < 2$
  - Pages les plus éditées :
    - $1 < k < 63$

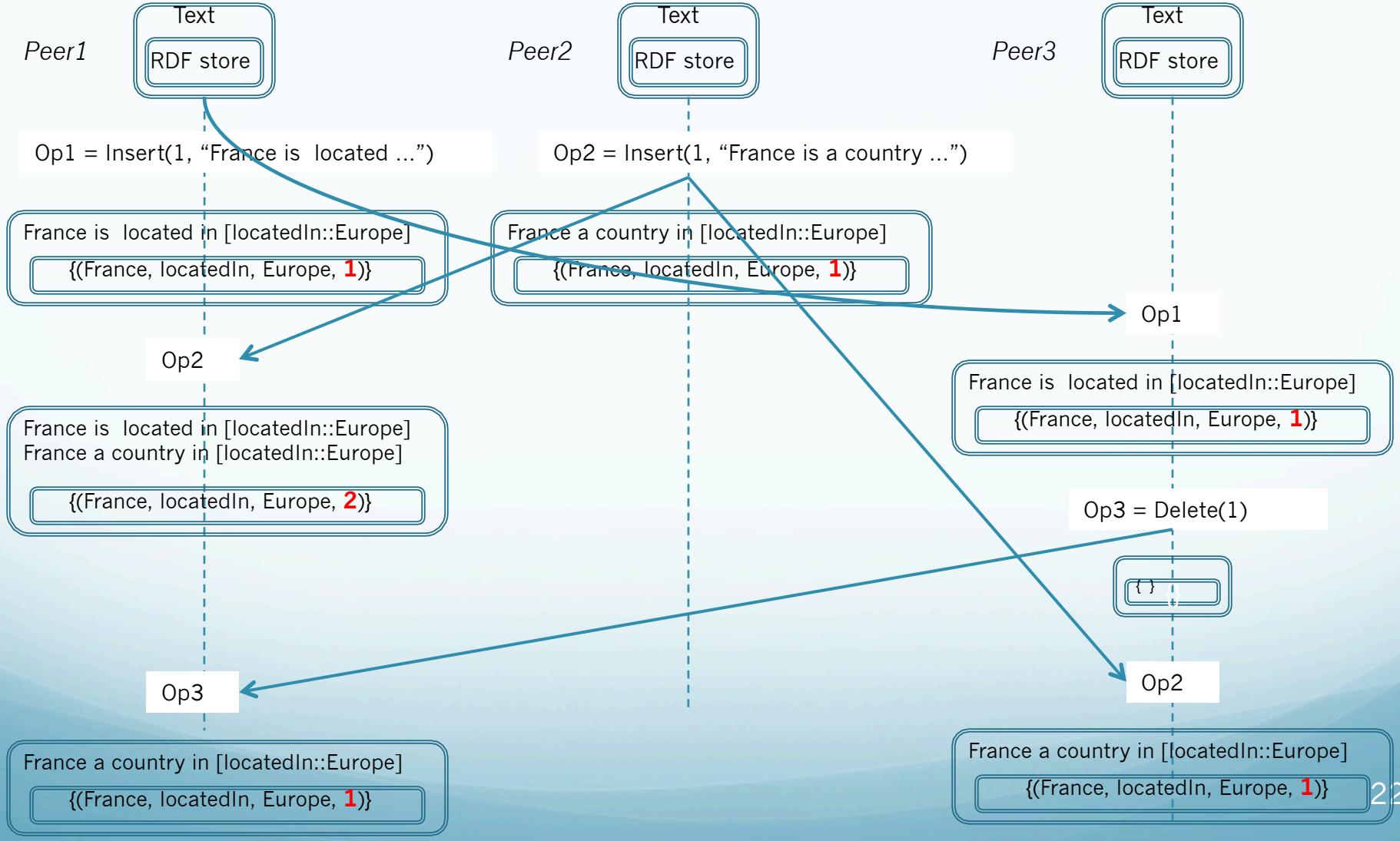
Logoot -> DSMW

- If text operations contain semantic annotation,
  - Update a semantic store by side effect of updating the text...
  - Require to redefine intentions of text operations to ensure convergence on text and semantic data...

Text + Annotation <> Text



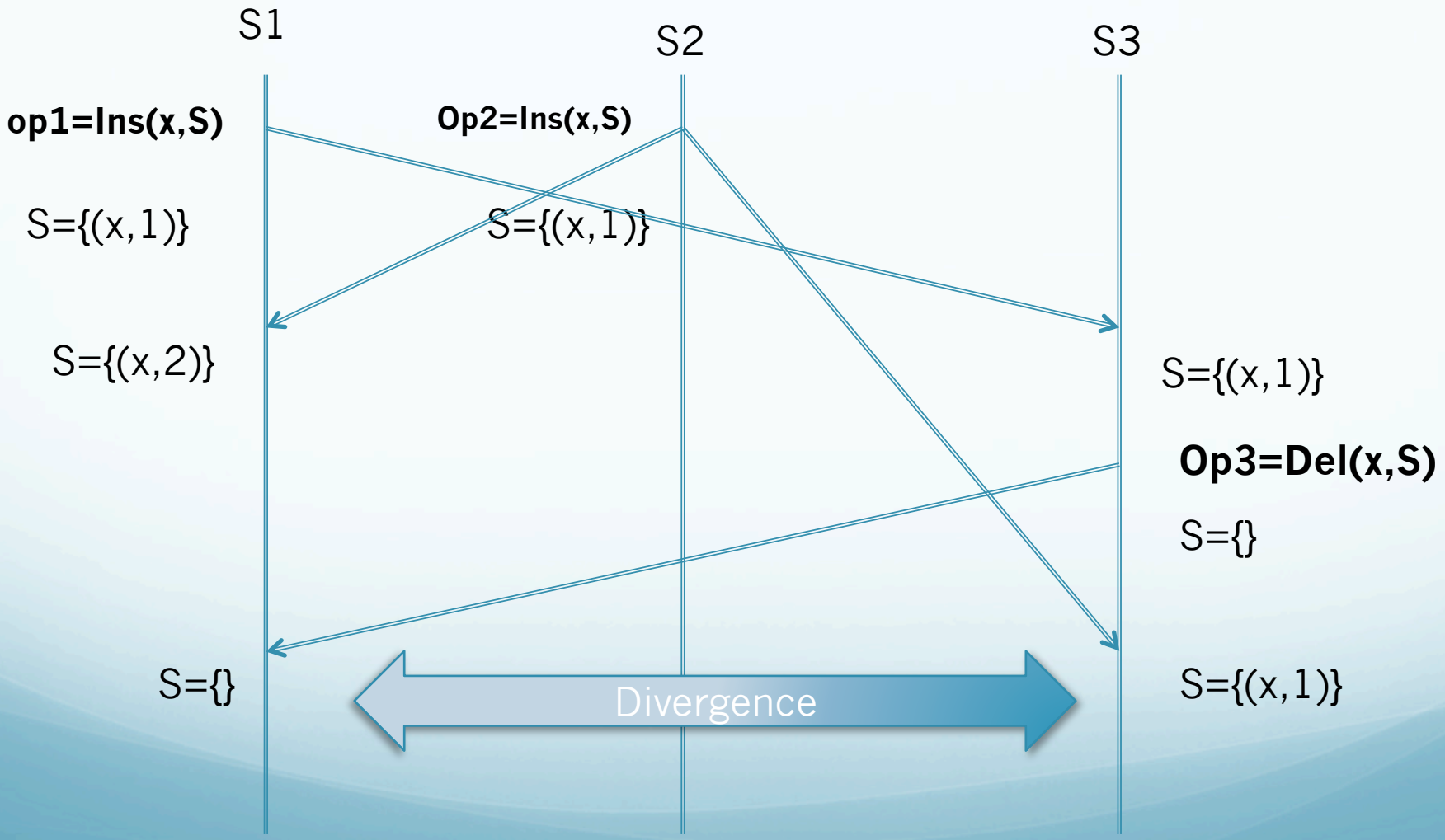
Adapt intentions of text operations



DSMW limitations…

- The semantic store is updated by side-effect.
  - Impossible to update semantic data without editing a text ! Not really a CRDT for semantic web…
- DSMW reify histories as semantic pages-> cost of provenance information, limitation for scalability
- DSMW has currently no UNDO support : cannot represent 'reject' operation in process support.
- From a CRDT point of view:
  - Composing CRDT should give a CRDT (commutativity)
  - So Logoot (CRDT for text) should composed with a CRDT for triples to obtain the CRDT text+data… But CRDT for triples does not exists… Why not just a multi-set ?

Because it does not work ☹



A proposal for sets

```
ins(e, S):
pre : ∄ k : (e,k) ∈ S.A ∧ k > 0

if (∃ k : (e,k) ∈ S.A ) then
 if (k≤ 0)
    S.A= (S.A / (e,k)) ∪ (e,1);
    send(rins((e,+|k|+1),S))
 else if (k>0) // if precondition not required
        S.A=(S.A /(e,k)) ∪ (e,k+1);
        send(rins((e,+1),S))
      endif
 endif
else
  S.A=S.A ∪ (e,1);
  send(rins((e,+1),S));
endif
```

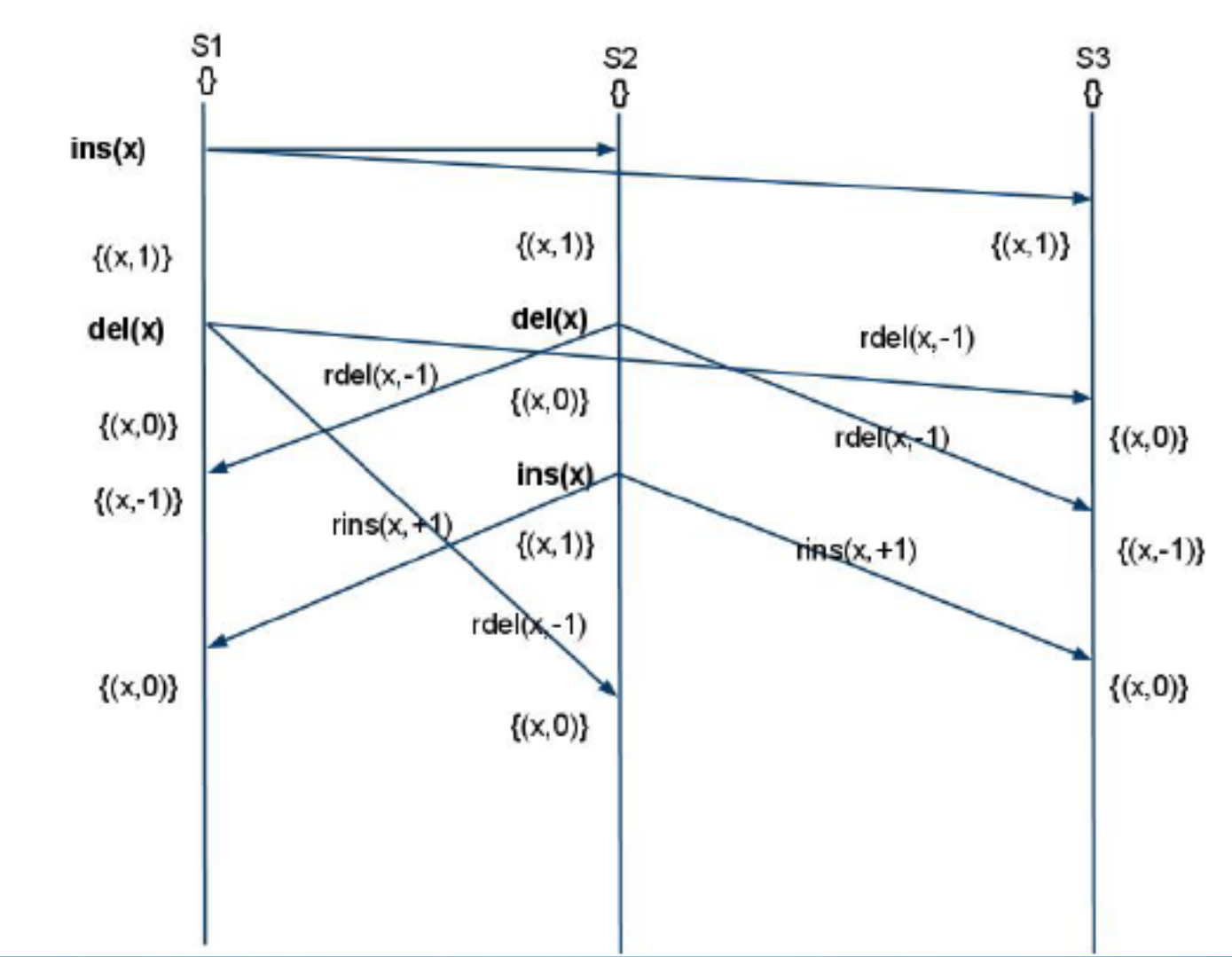
```
del(e, S):
//pre : ∃ k : (e,k) ∈ S.A ∧ k>0

if (∃ k : (e,k) ∈ S.A)
  if (k ≤ 0) then // If precondition is not required
     S.A=(S.A / (e,k)) ∪ (e,k−1);
     send(rdel((e,−1),S))
  else // k>0
    S.A=(S.A / (e,k)) ∪ (e,0);
    send(rdel((e,−k),S));
  endif
else // never happens if precondition is verified ,
     // but, IMHO, sending with 0 maybe crash something...
  S.A=S.A ∪ (e,0);
  send(rdel((e ,0), S));
endif
```

```
rdel((e, j : ℤ*), S):
pre : none

if (∃ k : (e,k) ∈ S.A)
  S.A=(S.A / (e,k)) ∪ (e,k+j)
else
  S.A=S.A ∪ (e,0+j)
endif
```

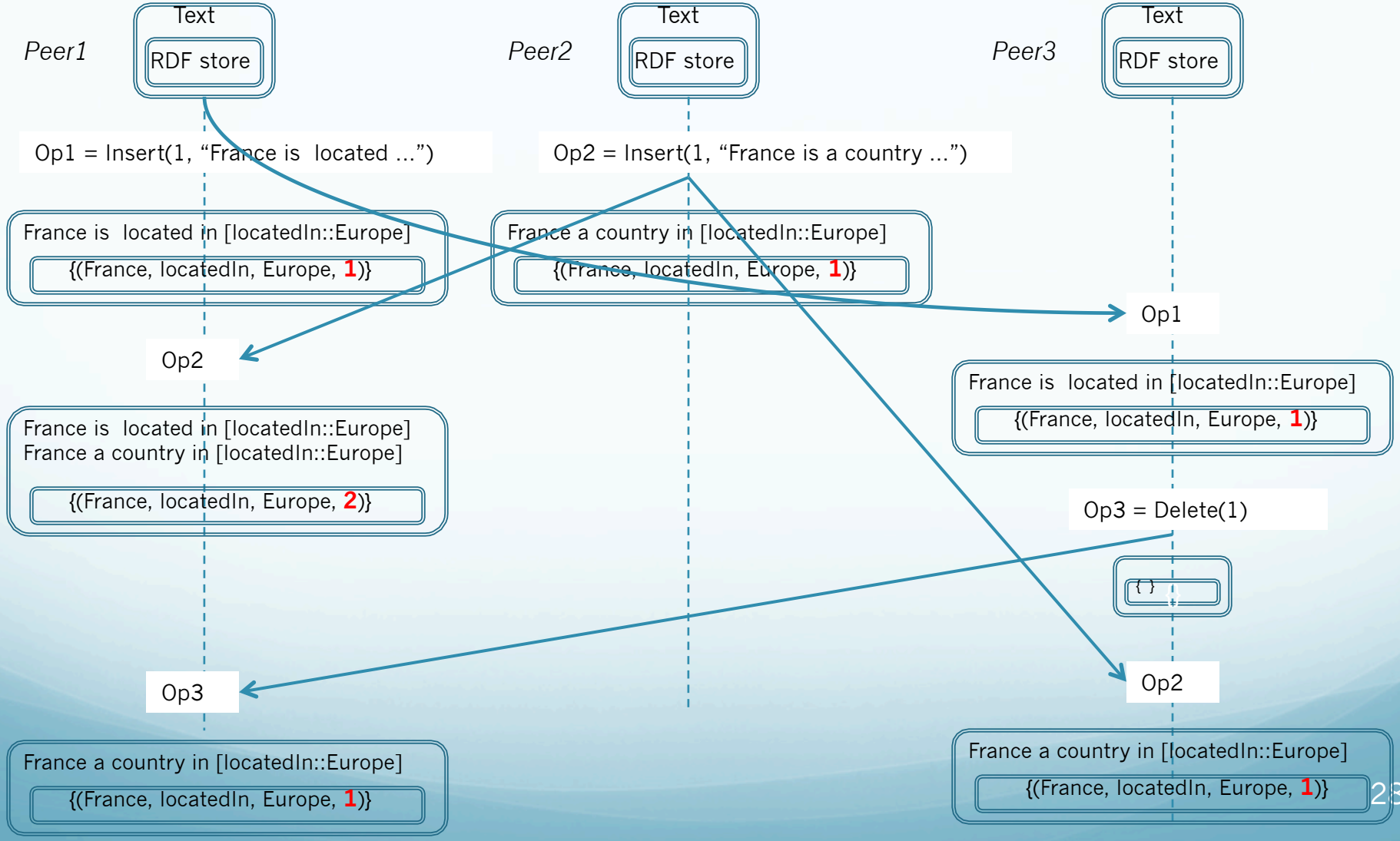
CRDTs for Semantic Store



Research Directions

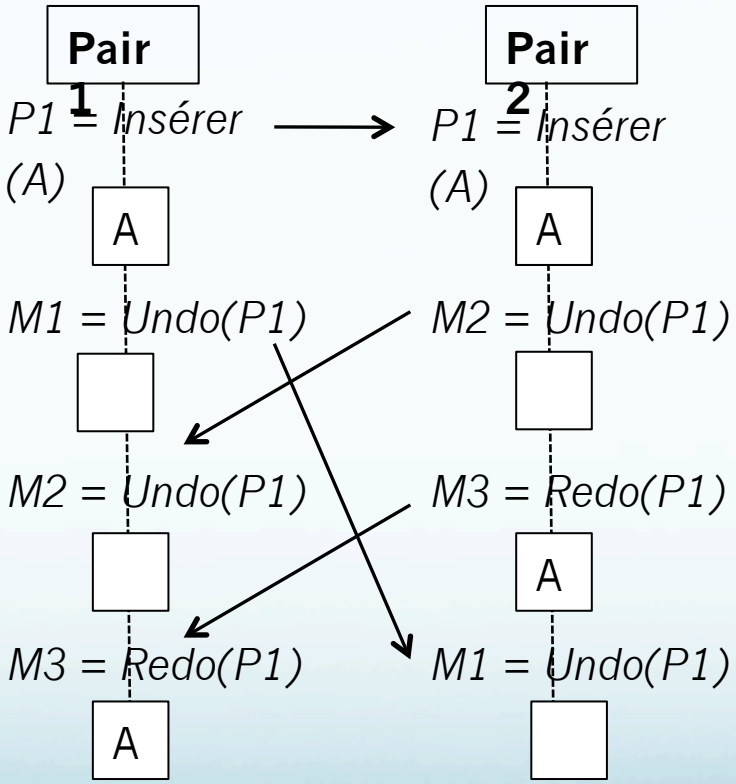
- Need to define and experiment a real CRDT just for semantic store.
- DSMW demonstrated how to embed a CRDT in real wiki, the challenge is to embed a specific CRDT in a an existing semantic store.
- Support for undo.
- Allow to build a network of semantic stores that are able to synchronize at any time. This store can be accessed by wiki or by any kind of tool.
- Need evaluations on real semantic datasets
- This is the goal of the proposed thesis

CRDT : Text -> semantic store



Swooki: Mécanisme d'annulation de groupe

- Dans ce mécanisme, tout utilisateur peut annuler à tout moment n'importe quelle opération d'édition.



- « Annuler une modification fait retourner le système à l'état qu'il aurait atteint si cette modification n'a jamais été produite. » [Sun02]

Distributed Semantic MediaWiki: CRDT in action

Pascal Molli
Full Professor, Nantes University
Head of GDD team, LINA,
pascal.molli@univ-nantes.fr
http://member.acm.org/~pascal.molli
Kickoff Concordant Tuesday, November 16, 2010

Context

- Social tools are evolving to social semantic tools to improve navigability, interoperability, and to allow exact search
  - Wikis are evolving to semantic wikis : Semantic Media Wiki, Kiwi
  - CMS are evolving to semantic CMS : Drupal 7
  - Forges are evolving to semantic forges : Qualipso, Helios

From Wikis to Semantic wikis

Content of wiki page of "France"	Content of semantic wiki page of "France"
France is located in [Europe]	France is located in [locatedIn::Europe]
The capital of France is [Paris]	The capital of France is [hasCapital:: Paris]

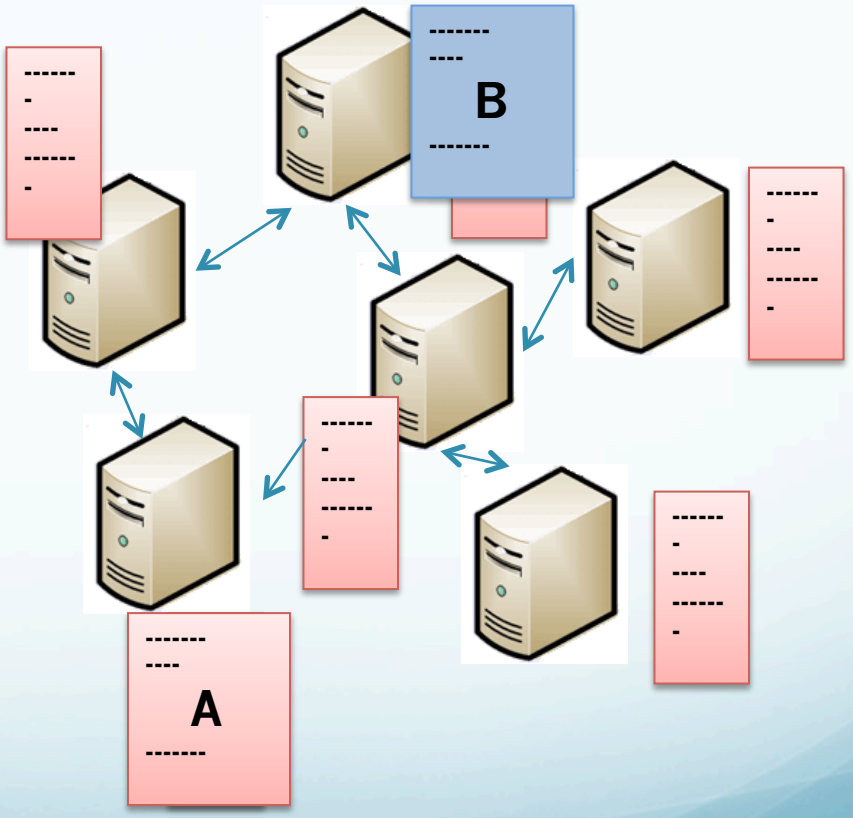
- Next, semantic queries can be performed:
  - [[locatedIn::Europe]][[hasCapital::Paris]]
  - Return the semantic wiki page « France ».

General issues

- Scalability in Social Semantic tools :
  - Mediawiki does not scale but Wikipedia scales, can a Semantic Wikipedia scale ?
- Content trust:
  - Content trust relies on lightweight coordination model in Wikipedia, does it still working for a semantic Wikipedia ?

DSMW Approach

- Instanciate the optimistic replication model for a semantic wiki.
  - Content replication allow to execute semantic queries on different hosts
  - Controling the propagation of changes in the network allows to represent editing processes for structured data



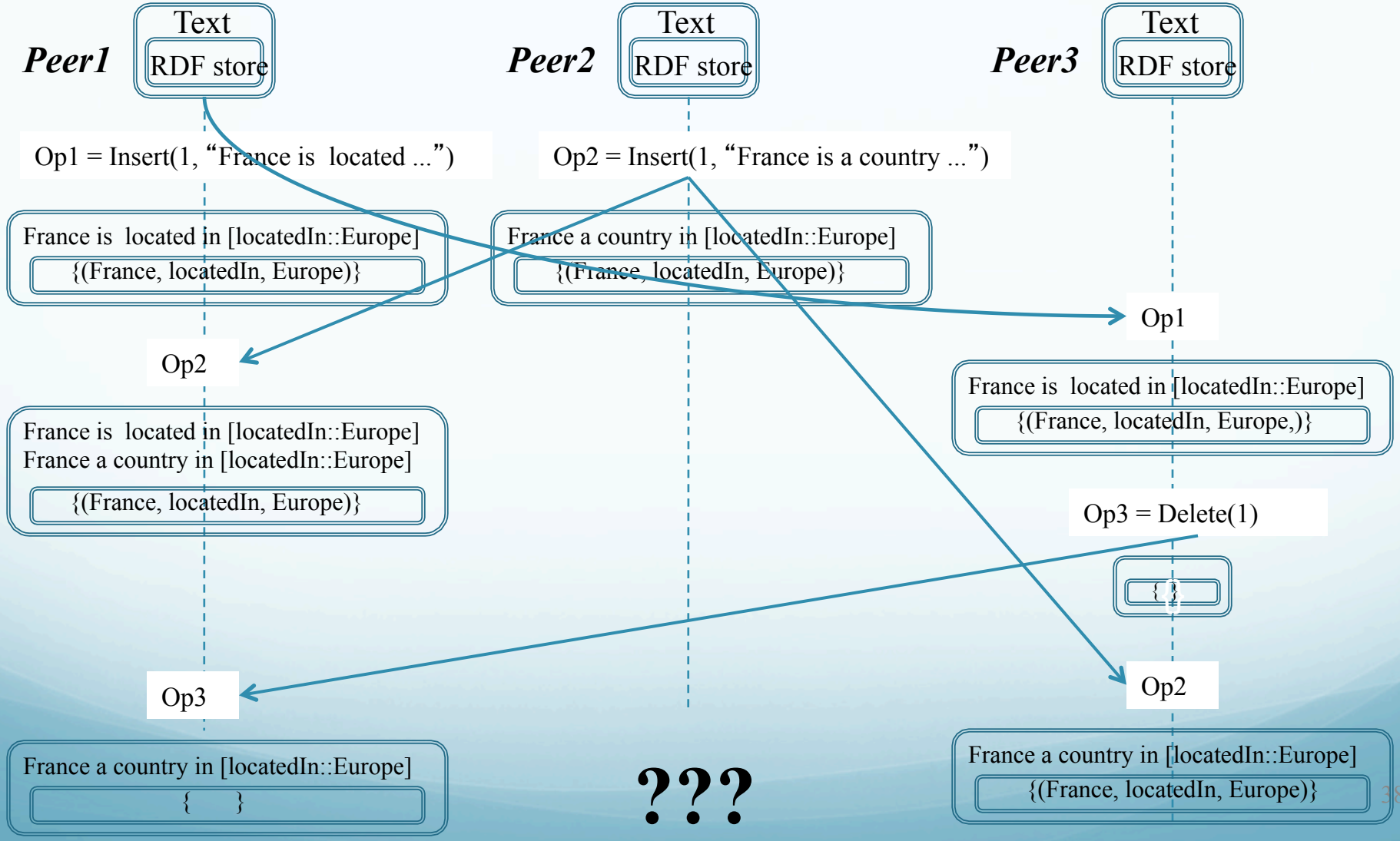
Related Scientific Problems

- Semantic wikis rely on semantic pages that mix text and semantic annotations
- Existing CRDT algorithms do not ensure Caualisaty, Convergence and Intentions on such data types.
- This raises the problem of genericity of CRDT algorithms compared to Operational Transformation approach.

Contribution

- We defined intention for new data type (text + semantic annotations)
- We extended the Logoot algorithm to handle this new data type
- We implemented the new algorithm as a pure Media Wiki plugin (DSMW).
  - Distributed in GPL on http://dsmw.org

Semantic pages and CCI consistency



Insert and delete intentions

- Intentions of op = Insert(IDPage,newLine,p, n) :

$\exists\, i\ i_N\ i_P\ .\ (i_P < i) \wedge \mathrm{LineID}(\mathrm{Page}(i_P)) = p \wedge (i \leq i_N) \wedge \mathrm{LineID}(\mathrm{Page}(i_N)) = n \Rightarrow$

$\mathrm{Page}'(i) = \mathrm{newline}$

$\wedge\ (\forall\, j.\ (j < i) \Rightarrow \mathrm{Page}'(j) = \mathrm{Page}(j))$

$\wedge\ (\forall\, j.\ (j \geq i) \Rightarrow \mathrm{Page}'(j) = \mathrm{Page}(j-1))$

$\wedge\ R' = R + T$

- Intentions of op = delete(IDPage, dl):

$\exists\, i.\ \mathrm{LineID}(\mathrm{Page}'(i)) = dl \wedge \mathrm{Visibility}(\mathrm{Page}(i)) \Rightarrow$

$\neg\mathrm{Visibility}(\mathrm{Page}'(i)) \wedge R' = R - T$

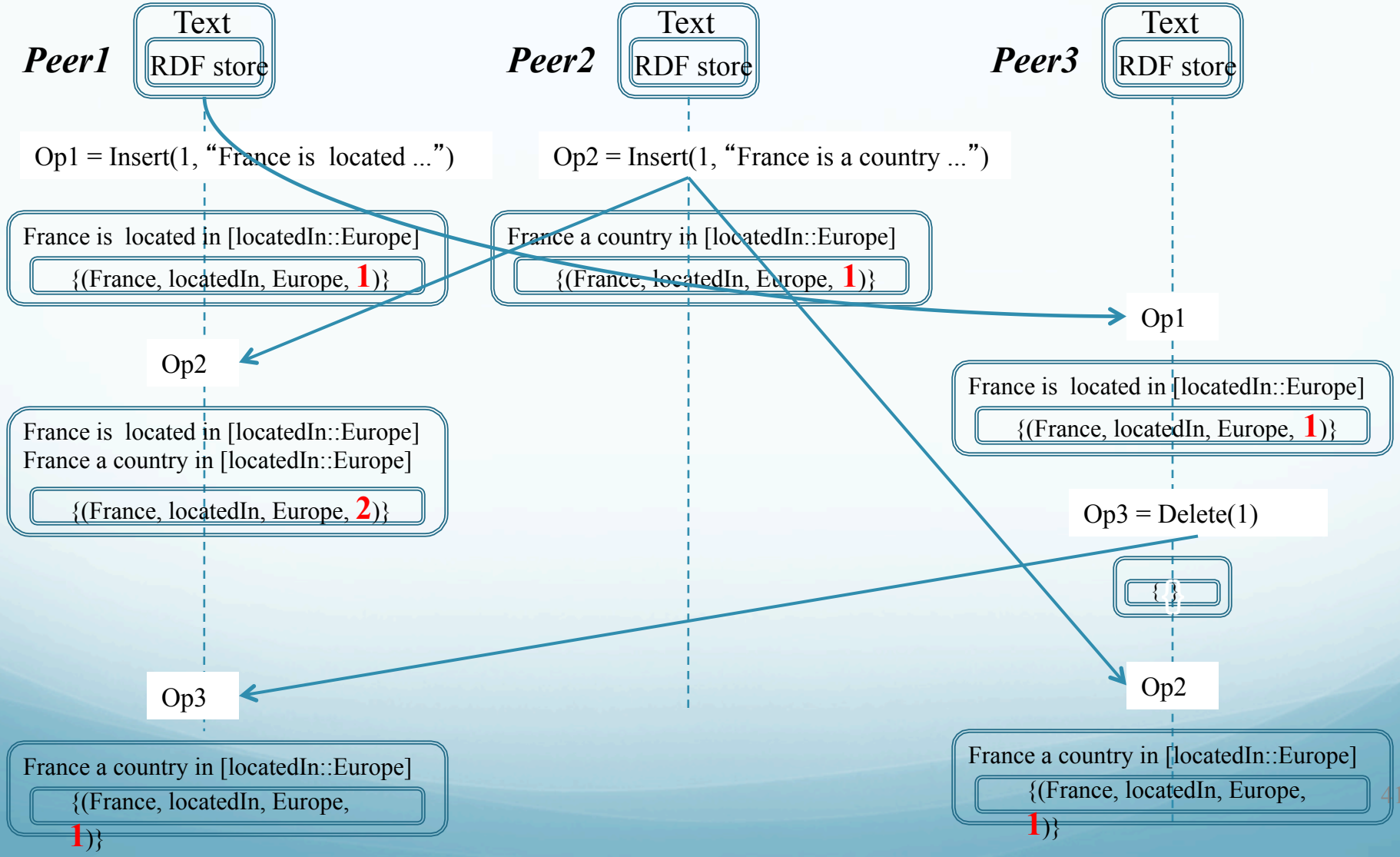
Algorithm modification

(on Woot for this slide)

```
IntegrateIns(PageID, line, l_P, l_N)
begin
let S' ← subseq(Page[PageID]), l_P, l_N)
if S = ∅ then
  insert(PageID, line, l_N)
else
      let i ← 0
      let d_min ← min(degree(S'))
      let F ← filter(S', degree = d_min)
      while (i < |F| − 1) and (F[i] <_id l) do
         i ← i +1
       IntegrateInsT(PageID, line, F[i−1], F[i])
let S ← ExtractRDF(line)
if S ≠ ∅ then
 for each triple ∈ S do
   if Contains(triple) then
       triple .counter++
   else
      insertRDF(R,triple)
   endif
endif
end
```

```
IntegrateDel(LineID)
begin
 Page[LineID]. visibility ← false
let S ← ExtractRDF(LineID)
 if S ≠ ∅ then
  for each triple ∈ S do
       triple .counter−−
     if triple .counter == 0 then
       deleteRDF(R,triple)
     endif
 endif
end
```

DSMW and CCI consistency



Demo (dsmw.ogv)

Log in / create account

page | discussion | view source | history | dsmw (5 patches)

navigation
- Main Page
- Download/Install
- Licensing:
- Getting Started
- FAQ
- Team
- Recent changes

documentation
- DSMW User Manual
- Presentation and Papers

community
- Support
- Contributing
- Issue tracker
- Forums
- Mailing lists
- News
- Roadmap

developpers
- Gforge Home
- Source documentation
- Source code
- DSMW Tests

Home

DSMW is an extension of Semantic Mediawiki (SMW). It allows to create a network of SMW servers that share common semantic wiki pages. DSMW manages synchronisation of shared semantic pages and ensures CCI consistency as in Google Wave. CCI stands for Causality, Convergence, Intentions (see papers for more informations). DSMW provides to SMW nearly the same features as a Distributed Version Control systems:

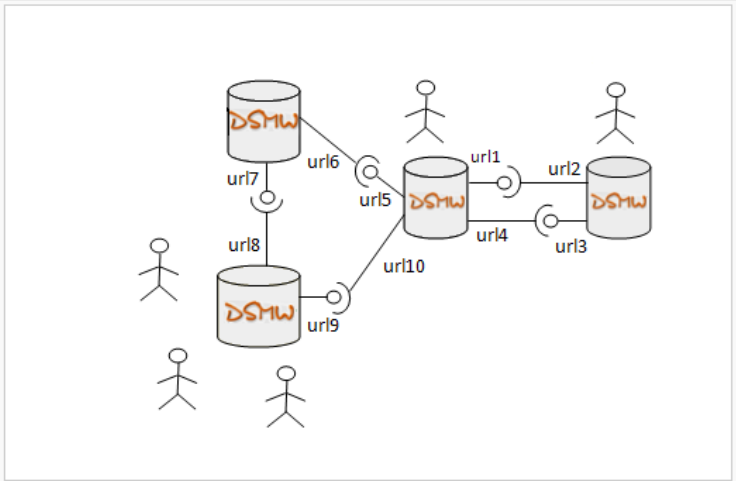
- you can work in isolation with your own server, test some stuff,
- publish changes to your own DSMW public feeds,
- you can also subscribe to any remote public DSMW feeds.

By this way, users can implements their own dataflows and represents any kind of dataflow oriented processes such as edit/review/publish.

DSMW is a network of a semantic wikis servers (DSMW peers). The number of peer is unknown. Semantic wiki pages can be replicated on several DSMW servers. A peer can replicate all or just some semantic wiki pages. The communication between DSMW peers is made through channels (feeds). This feeds are transporting operations that, when applied, modify local semantic wiki pages. If you are common to version control systems, a DSMW peer can be seen as a workspace, and a feed can be seen as a branch. DSMW works as follow:

- When a semantic wiki page is updated on a DSMW server, it generates a set of corresponding operations (Patch).
- The user can publish this patch into channels (PushFeed). The publication mechanism uses a semantic query. This query defines the pages' patches to be published (pushed).
- An authorized server can pull the patch(es) and integrated it (them) to the local replica of the pages. If needed, the integration process merges this modification with concurrent ones, generated either locally or received from a remote server. To do this, the user must create a PullFeed with the corresponding PushFeed features. For more information see (DSMW User Manual).

The system is correct if it ensure Causality, Convergence and Intention Preservations. Generation and integration is managed by the Logoot algorithm (see papers). DSMW allows users to build their own cooperative networks. The construction of the collaborative community is declarative, in the sense, every user declares explicitly with whom she would like to cooperate. A user can have her DSMW server installed on her machine. She can create and edit her own semantic wiki pages as in a normal semantic wiki system.



DSMW Network

Perspectives

- DSMW Experimentation will be done in ANR KOLFLOW 2010-2013 and in TAAABLE contest 2011.
  - Corpus will be expressed in DSMW ontology.
  - Can be used in ANR CONCORDANT and ANR STREAMS
- Open issues for CRDT :
  - Genericity in CRDT, composition of CRDT
  - User Acceptability of CRDT behaviors