

4^{ème} Meeting – Kolflow Team Silex

20-21 septembre 2012

Plan

- Assistance
 - Assistance à la tâche
 - Assistance à la résolution de conflit
 - Scenario SILEX
 - Principe de l'assistance
 - Quid de l'implémentation de K-CIP ?
 - Quid d'un prototype commun ?
- Délivrables Silex T+18
 - 4.4: Test and evaluation of the alter ego assistant with regard to the scenarios
 - 5.3: Algorithms and architecture for a distributed semantic wiki and a distributed alter-ego assistant

Types d'assistance

- 2 types d'assistance
 - À la tâche
 - Pour les nouveaux utilisateurs de DSMW
 - Fournie sur demande de l'utilisateur
 - Assistance
 - Guide l'utilisateur pour finir sa tâche
 - Finit la tâche de l'utilisateur à sa place
 - À la résolution de conflits
 - Quand un utilisateur intègre dans son wiki les données d'un autre wiki
 - Principe définit en collaboration avec la tâche 3

Assistance à la tâche


- Utilisation classique d'un wiki :
 - 1. Comment créer une page ?
 - 2. Comment modifier une page ?
 - 3. Comment trouver une page existante (recherche) ?
 - 4. Comment mettre des commentaires sur une page (onglet discussion) ?
-
- Concernant le partage :
 - 5. Comment mettre à disposition / partager une page ?
 - 5a. Comment créer un canal (faire un PushFeed) ?
 - 5b. comment mettre à jour le partage (faire un Push) ?
 - 6. Comment récupérer le contenu d'une page d'un autre wiki ?
 - 6a. Comment créer un canal (faire un PullFeed) ?
 - 6b. comment mettre à jour faire un Pull
-
- Visualisation de différence :
 - 7. Comment visualiser les changements que l'on fait sur une page avant de valider ?
 - 8. Comment visualiser l'historique d'une page ?
 - 9. Comment voir ce qui va être fait lors de la récupération des données d'un autre wiki ?
-
- Bonus :
 - toutes les questions liés aux boutons de Collectra
 - toutes les questions liés aux boutons d'Assistance
 - toutes les tâches de maintenance

Assistance à la tâche

- Pour chaque question, on propose :
 - une réponse explicative textuelle avec éventuellement des schémas (surtout pour le partage)
 - une vidéo des actions à faire
 - des exemples de traces
 - de faire la tâche pour l'utilisateur
-
- Pour trouver les traces servant d'exemples :
 - Définir les signatures des différentes tâches
 - Lister les actions sur le wiki
 - Lister les obsels correspondants aux actions dans les traces
 - Définir la requête permettant de récupérer tous les obsels dans le kTBS
 - Rechercher dans ses propres traces ou dans celles des autres (importation de traces)

Assistance à la tâche

- Le système dispose des signatures :
 - Création de page
 - Edition de page
 - Recherche
 - Discussion
 - PushFeed
 - PullFeed
 - Push
 - Pull
 - Historique



Ces signatures
ont été définies
par des experts

Exemple : création d'une page

- Quatre possibilités (au moins) :

Listes des actions pour la possibilité A :

- A1. Saisir le nom de la page dans la zone de recherche
- A2. Valider avec le bouton Go
- A3. Si la page existe, recommencer avec un autre nom
- A4. Si la page n'existe pas, cliquer sur le lien du nom de la page
- A5. Saisir du texte dans la zone de saisie
- A6. Cliquer sur le bouton "Save page »

Listes des actions pour la possibilité B :

- B1. Saisir le nom de la page dans la zone de recherche
- B2. Valider en tapant Entrée sur le clavier
- B3. Si la page existe, recommencer avec un autre nom
- B4. Si la page n'existe pas, cliquer sur le lien du nom de la page
- B5. Saisir du texte dans la zone de saisie
- B6. Cliquer sur le bouton "Save page »

Listes des actions pour la possibilité C :

- C1. Dans une page en édition, ajouter un lien avec le bouton dédié de l'interface
- C2. Eventuellement saisir du texte dans la zone de saisie
- C3. Cliquer sur le bouton "Save page" de la page en cours
- C4. Cliquer sur le lien de la page à créer
- C5. Saisir du texte
- C6. Cliquer sur le bouton "Save page" de la page à créer

Listes des actions pour la possibilité D :

- D1. Dans une page en édition, saisir avec le clavier un lien
- D2. Eventuellement saisir du texte dans la zone de saisie
- D3. Cliquer sur le bouton "Save page" de la page en cours
- D4. Cliquer sur le lien de la page à créer
- D5. Saisir du texte
- D6. Cliquer sur le bouton "Save page" de la page à créer

Exemple : création d'une page

- Quatre possibilités (au moins) :

Listes des actions pour la possibilité A :

- A1. Saisir le nom de la page dans la zone de recherche
- A2. Valider avec le bouton Go
- A3. Si la page existe, recommencer avec un autre nom
- A4. Si la page n'existe pas, cliquer sur le lien du nom de la page
- A5. Saisir du texte dans la zone de saisie
- A6. Cliquer sur le bouton "Save page »

Listes des obsels correspondants :

- A1. Obsels de type ChangeText, attribut title = "changed searchInput"
- A2. Obsel MouseEvent, attribut element_id = "SearchGo Button"
- A3. à voir
- A4. Obsel MouseEvent, attribut element_href = "http://localhost/wiki2/index.php?title=NomPage&action=edit&redlink=1"
en remplaçant l'adresse du wiki2 par la bonne adresse
et en remplaçant NomPage par le nom saisi par l'utilisateur.
Pour trouver le nom de la page, il faut récupérer dans la trace A1, les noms de pages (car il y a toutes les saisies...) => nom de la page dans l'attribut Changes (la partie non barrée) du dernier ChangeText avant chaque MouseEvent - SearchGo
- A5. Si on décide de montrer un exemple de saisie, il faut récupérer toutes les actions faites dans la zone de saisie (KeyEvent, ChangeText, clic sur les boutons de mise en page...) donc toutes les actions entre la date de fin de la trace A4 et la date de début de la trace A6.
- A6. Obsel MouseEvent, attribut location = NomPage avec le chemin complet, attribut element_id = wpSave

Exemple : création de page

```
@prefix ktbs: <http://liris.cnrs.fr/silex/2009/ktbs#>.
@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfrest: <http://liris.cnrs.fr/silex/2009/rdfrest#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix xml: <http://www.w3.org/XML/1998/namespace>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
```

```
<> ktbs:contains <Trace_CreatePageLink9/>.
```

```
<Trace_CreatePageLink9/> a ktbs:ComputedTrace;
  ktbs:hasParameter ""sparql=
```

```
PREFIX : <http://localhost:8001/base1/modell1/>
PREFIX ktbs: <http://liris.cnrs.fr/silex/2009/ktbs#>
```

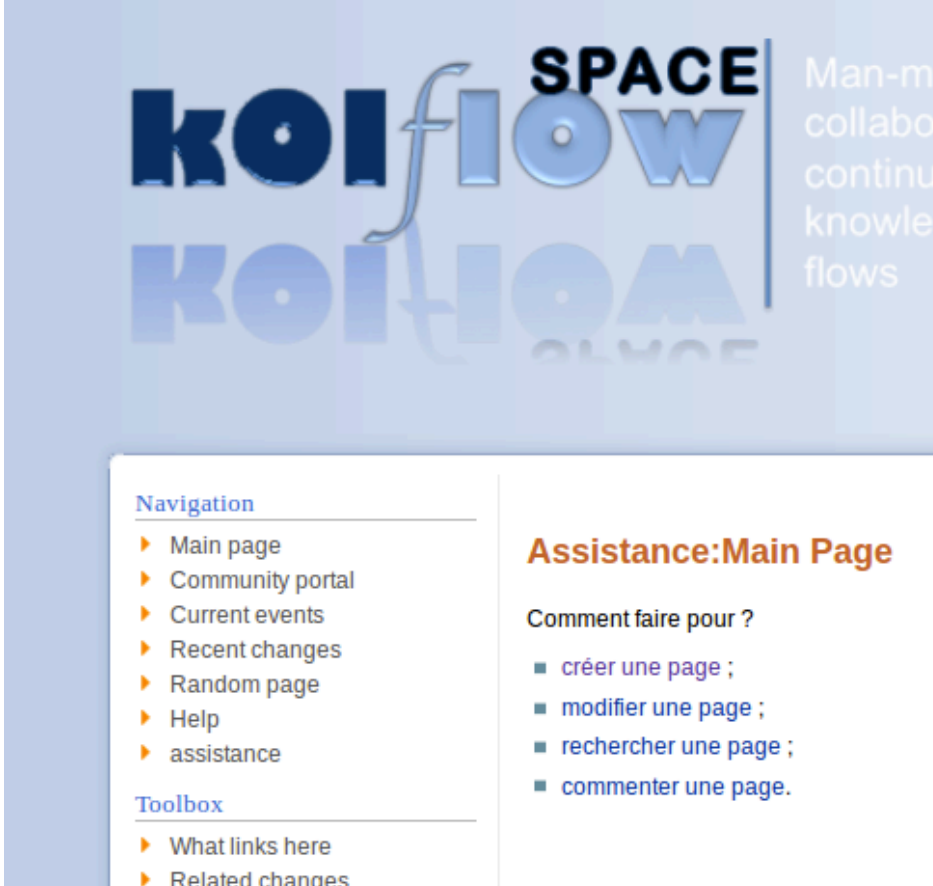
```
CONSTRUCT {
  [ a :MouseEvent ;
    ktbs:hasTrace <%(__destination__)s> ;
    ktbs:hasBegin ?begin ;
    ktbs:hasBeginDT ?begindt ;
    ktbs:hasEnd ?end ;
    ktbs:hasEndDT ?enddt ;
    :user_id ?user_id ;
    :user_name ?user_name ;
    :site_id ?site_id ;
    :title ?title ;
    :description ?description ;
    :location ?location ;
    :element_type ?element_type ;
    :element_id ?element_id ;
    :element_value ?element_value ;
    :element_text ?element_text ;
    :element_checked ?element_checked ;
    :element_src ?element_src ;
    :element_href ?element_href ;
    ktbs:hasSourceObsel ?obsel ;
  ] .
```

```
} WHERE {
  ?obsel a :MouseEvent ;
    ktbs:hasBegin ?begin ;
    ktbs:hasBeginDT ?begindt ;
    ktbs:hasEnd ?end ;
    ktbs:hasEndDT ?enddt ;
    :user_id ?user_id ;
    :user_name ?user_name ;
    :site_id ?site_id ;
    :title ?title ;
    :description ?description ;
    :location ?location ;
    :element_type ?element_type ;
    :element_id ?element_id ;
    :element_value ?element_value ;
    :element_text ?element_text ;
    :element_checked ?element_checked ;
    :element_src ?element_src ;
    :element_href ?element_href.
  FILTER regex(?element_href, "&action=edit&redlink=1", "i") .
}""";
  ktbs:hasMethod ktbs:sparql;
  ktbs:hasSource <http://localhost:8001/base1/PrivateTrace_0/>.
```

X requêtes + une requête de fusion

Assistance à la tâche - Bilan

- On a :
 - Un collecteur - Collectra
 - Des traces dans le kTBS
 - Traces personnelles
 - Traces collectives
 - Traces importées
 - La majorité des requêtes sparql
 - Une interface intégrée à DSMW (pages wiki)
 - Texte explicatif
 - Lancement de la requête et récupération des traces
 - Une automatisation de l'action de push-pull
- Mais :
 - Pas de vidéos
 - Affichage des traces sous forme de tableau
 - Pas d'automatisation de toutes les tâches



The screenshot displays the 'KOIFLOW SPACE' interface. The header features the logo 'KOIFLOW SPACE' in blue and black, with the tagline 'Man-m collaboration continue knowledge flows' on the right. Below the header, there are two main sections: 'Navigation' and 'Assistance:Main Page'. The 'Navigation' section lists links for 'Main page', 'Community portal', 'Current events', 'Recent changes', 'Random page', 'Help', and 'assistance'. The 'Assistance:Main Page' section is titled 'Comment faire pour ?' and lists four actions: 'créer une page ;', 'modifier une page ;', 'rechercher une page ;', and 'commenter une page.'.

KOIFLOW SPACE Man-m collaboration continue knowledge flows

Navigation

- ▶ [Main page](#)
- ▶ [Community portal](#)
- ▶ [Current events](#)
- ▶ [Recent changes](#)
- ▶ [Random page](#)
- ▶ [Help](#)
- ▶ [assistance](#)

Assistance:Main Page

Comment faire pour ?

- [créer une page ;](#)
- [modifier une page ;](#)
- [rechercher une page ;](#)
- [commenter une page.](#)

Toolbox

- ▶ [What links here](#)
- ▶ [Related changes](#)

Plan

- Assistance
 - Assistance à la tâche
 - Assistance à la résolution de conflit
 - Scenario SILEX
 - Principe de l'assistance
 - Quid de l'implémentation de K-CIP ?
 - Quid d'un prototype commun ?
- Délivrables Silex T+18
 - 4.4: Test and evaluation of the alter ego assistant with regard to the scenarios
 - 5.3: Algorithms and architecture for a distributed semantic wiki and a distributed alter-ego assistant

Vision de Silex

sur la gestion de conflits

1. Woody a une instance de WikiTaaable sur son wiki avec les ontologies (ingrédients, aliments, etc.) et la base de recettes.
2. Woody veut une recette d'omelette aux asperges soumet la requête RRw "plat principal + asperge" au système Taaable utilisant son instance de WikiTaaable.
3. Woody ne reçoit aucune recette qui lui convienne.
4. Woody explore les recettes identifie la recette R1 "omelette aux oignons" pense que cette recette pourrait être adaptée pour devenir une recette d'omelette aux asperges.
5. Woody change la catégorie de l'ingrédient Asperge pour la passer de "Légume" à "Oignon" dans le but que le système lui propose une adaptation de la recette R1.
6. Woody vérifie ses hypothèses en resoumettant le requête RRw
7. Taaable lui renvoie la recette R1w une adaptation faite de la recette R1.
8. Woody a ce qu'il veut, il est content :-)

Vision de Silex

sur la gestion de conflits

9. Woody lance les tests locaux pour vérifier que tout fonctionne encore avec sa modification.
10. Les résultats sur son wiki sont bons.
11. Woody publie sa modification.
12. Sarah et Max récupèrent cette modification lancent les tests sur leurs wikis respectifs.
13. Les tests locaux sont bons.
14. Sarah et Max disent à Woody que sa modification est une bonne idée par rapport aux possibilités d'adaptation qu'il doit la soumettre à la communauté.
15. Woody publie sa modification (pour rappel le changement de catégorie de l'ingrédient Asperge) au modérateur de la communauté.
16. Le lendemain, le modérateur s'intéresse aux "push" du jour.
17. Le modérateur récupère la proposition de Woody.
18. Le modérateur lance un outil de vérification de non régression.

Vision de Silex sur la gestion de conflits

19. Résultat

X recettes {Rx} du jeu de test {Rz} appartenant à {R+} ne sont plus retrouvés (cf. papier K-CIP).

20. Le modérateur lance l'outil d'assistance à la gestion des conflits / ontologie.

21. L'outil d'assistance propose la création d'un groupe de négociation de sens.

22. Pour constituer ce groupe, il considère :

la personne ayant fait la proposition : Woody

les créateurs et éditeurs des recettes Rx mises en causes : {Experts-Rx}

Bonus : à partir des traces des wikis de la communauté

identifier les utilisateurs des recettes Rx pour intégrer le groupe.

23. Chacune des personnes acceptant de faire partie du groupe de négociation accepte également de partager ses traces.

Vision de Silex sur la gestion de conflits

24. Ensuite processus itératif
à partir des traces de Woody
retrouver les personnes ayant récupérées la catégorie Asperge
de l'ontologie de Woody
sans faire de modification
(ce qui peut laisser penser qu'ils sont d'accord avec la proposition).
25. Pour chacune de ces personnes (ici, Sarah et Max)
le processus va chercher dans leurs traces
des personnes ayant récupérer la modification sans la remettre en question
et ainsi de suite.

=> Le groupe créé contiendra donc potentiellement :

la personne soumettant la proposition : {Woody}

la sous-communauté ayant "validée" sa proposition : {Sarah et Max}

les experts des recettes mises en cause Rx :

les créateurs et éditeurs des recettes i.e. {Experts-Rx}

éventuellement les utilisateurs des recettes Rx : {Users-Rx}

le modérateur {M}

Vision de Silex sur la gestion de conflits

26. L'outil d'assistance proposera ensuite aux membres du groupe une page de négociation avec :
 1. récupération des traces individuelles
relatives à l'édition des ressources concernées
(recettes Rx et ontologie des ingrédients)
 2. récupération des actions collectives sur ces ressources

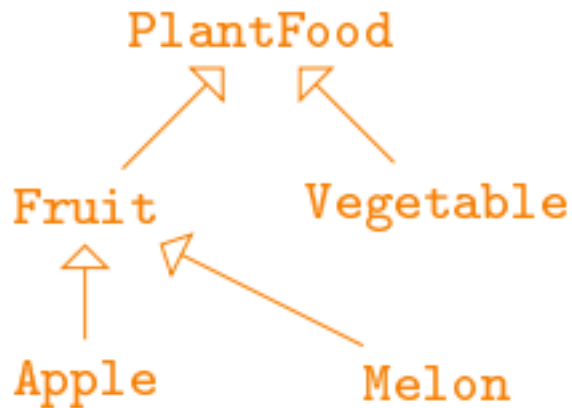
Détection de conflits sémantiques

- *Ce que j'ai compris des explications de Jean ;-P*
 - Lors de l'importation des données d'un wiki dans un autre wiki
- => 2 possibilités :
- Fusion de 2..n ontologies
 - Révision d'une ontologie par 1..n autres

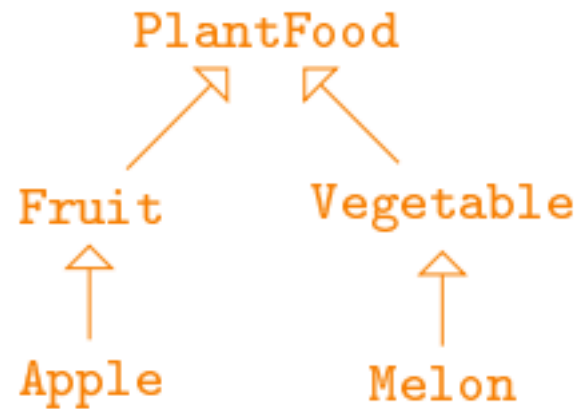
Ontologies

- En fait, ce sont des Taxonomies : $A \sqsubseteq B$

- Ψ_1



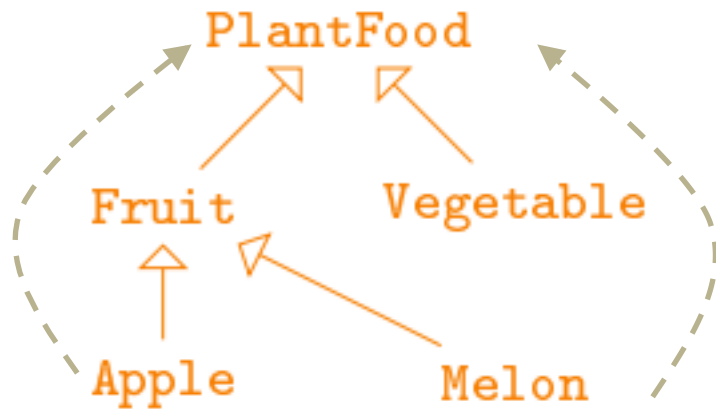
- Ψ_2



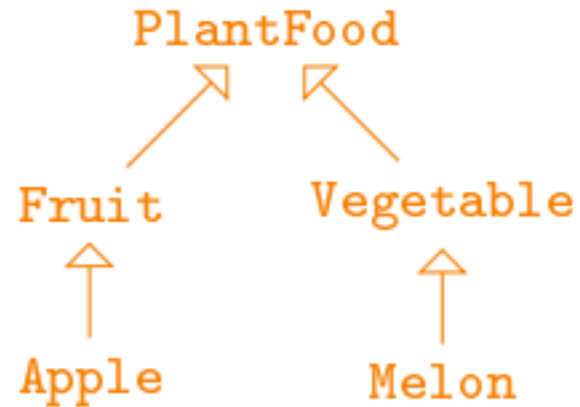
Fusion de deux taxonomies

- 1ère étape : fermeture déductive

- ψ_1



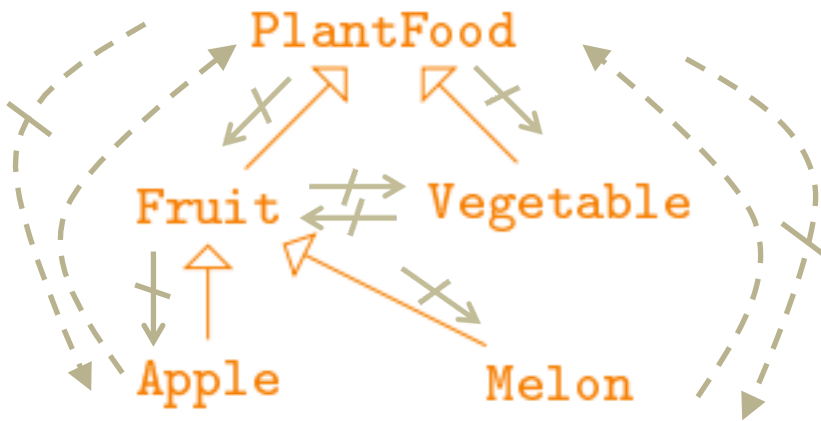
- ψ_2



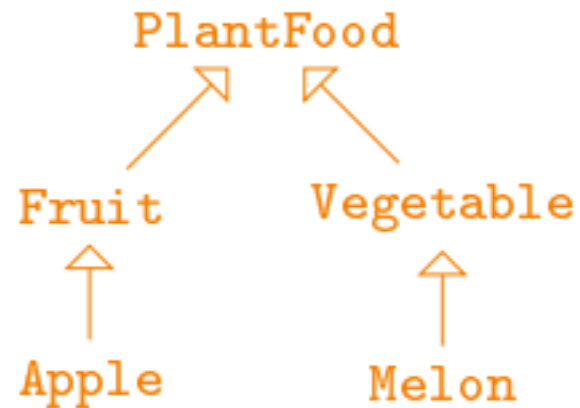
Fusion de deux taxonomies

- 2ème étape : non-subsumption
 - On rajoute explicitement tous les liens qui n'existent pas

• Ψ_1



ψ_2



Algo de fusion

Cf. livrable 3.1.

1. On fait la fermeture déductive des deux taxonomies

$$\psi_1 \longrightarrow \widehat{\psi}_1 \qquad \psi_2 \longrightarrow \widehat{\psi}_2$$

2. On garde ce qui est commun

$$\alpha = \widehat{\psi}_1 \cap \widehat{\psi}_2$$

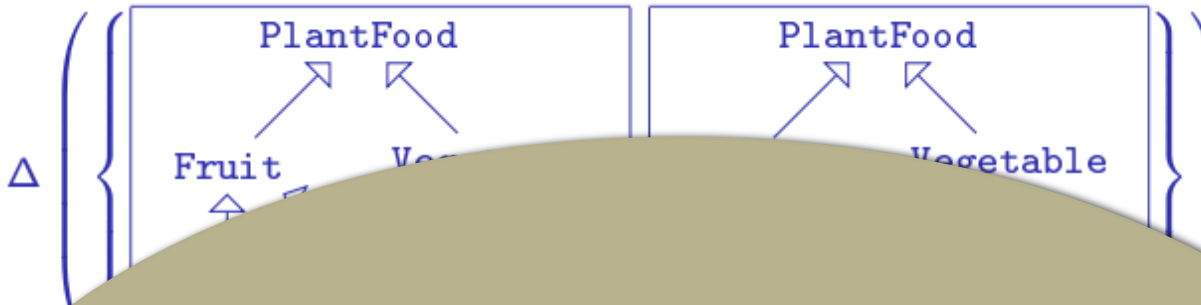
3. On s'intéresse à ce qui est différent

$$\delta_{1/2} = \widehat{\psi}_1 \setminus \widehat{\psi}_2$$

on prend le maximum de $\alpha \cup \delta$ qui est cohérent

=> on a 4 possibilités

Résultat de la fusion



**Qui décide de la
possibilité que l'on
garde ?**

On supprime
les conflits
Melon n
Fruit, n
Vegeta
sous P

On garde

On ne garde
que la droite

On ne
«
des deux
taxonomies

Apple

Melon

Concrètement qu'est-ce qui se passe quand on utilise DSMW ?

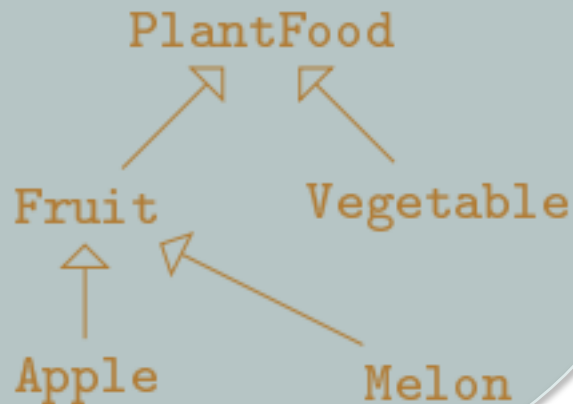
On a 2 wikis identiques et des outils :

Δ : boîte noire de fusion (ou de révision)

T : boîte noire de tests

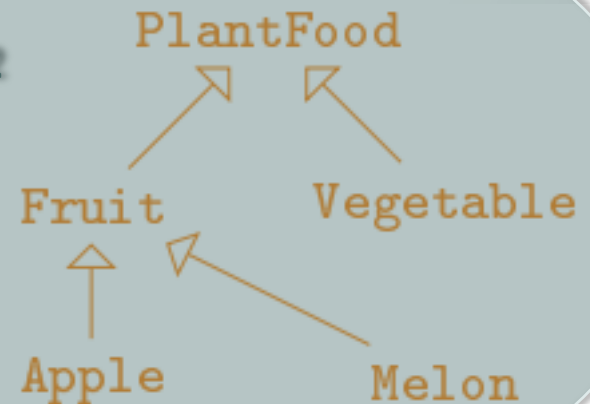


WIKI 1



$\psi_1 =$

WIKI 2



$\psi_2 =$

Concrètement qu'est-ce qui se passe quand on utilise DSMW ?

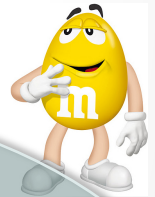
Quand Mr Jaune modifie son ontologie des ingrédients pour mettre *Melon* non plus sous *Fruit* mais sous *Vegetable*

T

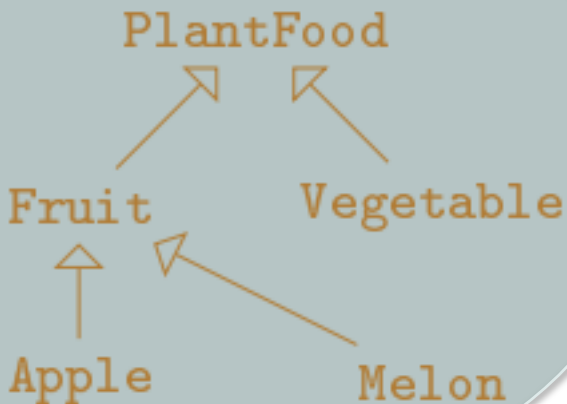
Δ

=> Création d'un patch

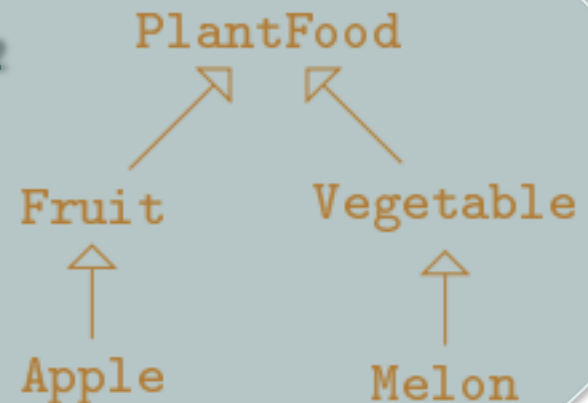
-- Melon \sqsubseteq Fruit
+ Melon \sqsubseteq Vegetable



WIKI 1



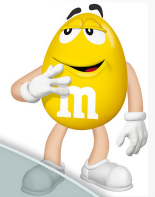
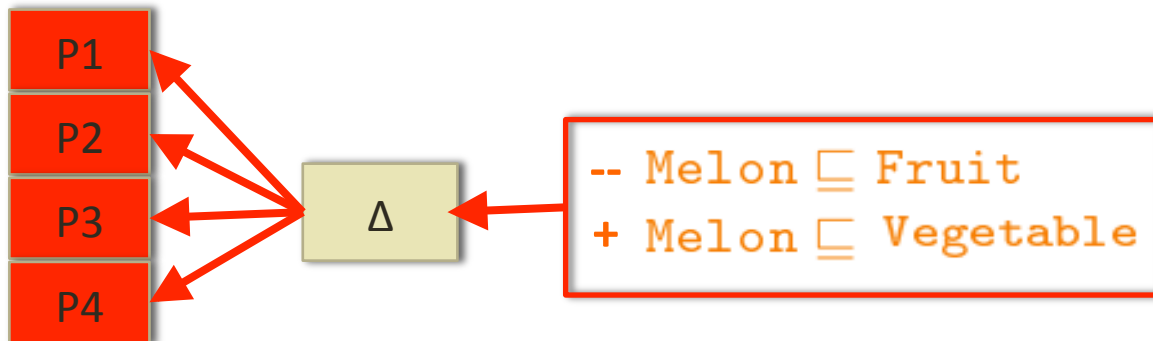
WIKI 2



Concrètement qu'est-ce qui se passe quand on utilise DSMW ?

Quand Mr Jaune partage son wiki et que Mr Rouge récupère, il se passe..

T



WIKI 1

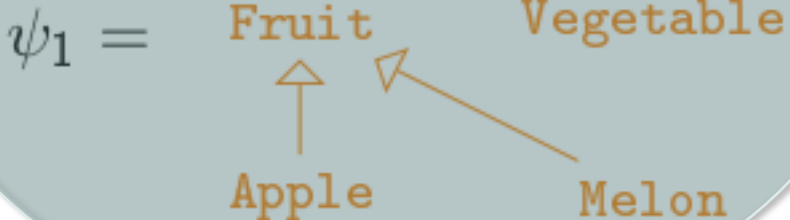
PlantFood

Fruit

Vegetable

Apple

Melon



WIKI 2

PlantFood

Fruit

Vegetable

Apple

Melon

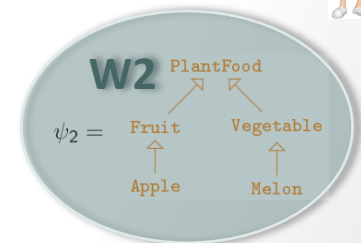
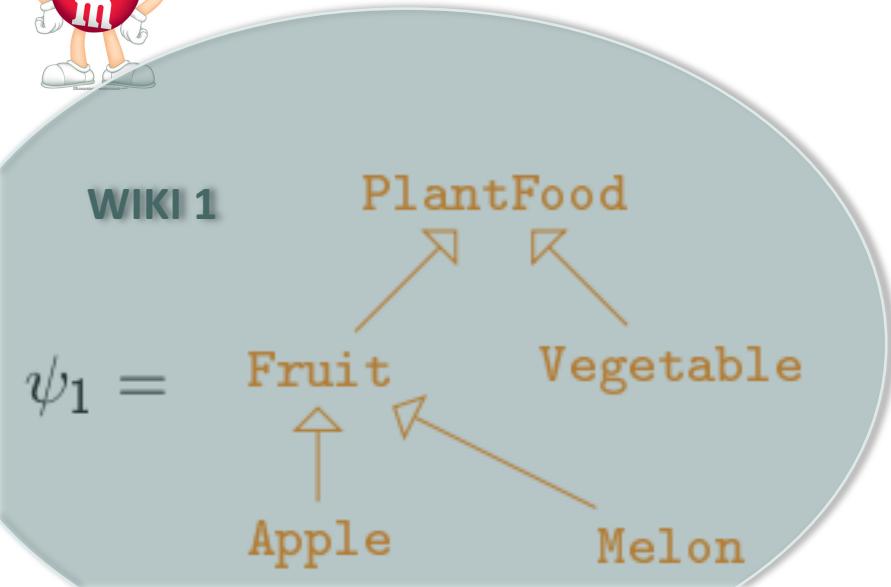
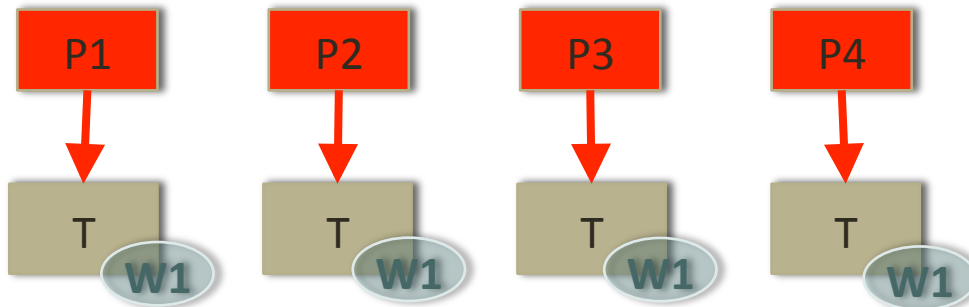


Concrètement qu'est-ce qui se passe quand on utilise DSMW ?

Quand Mr Jaune partage son wiki et que Mr Rouge récupère, il se passe..

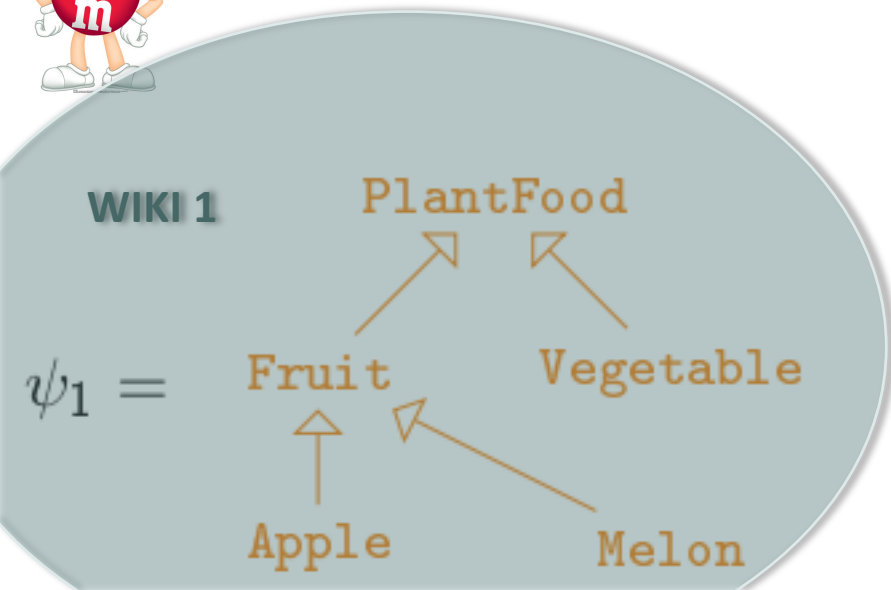
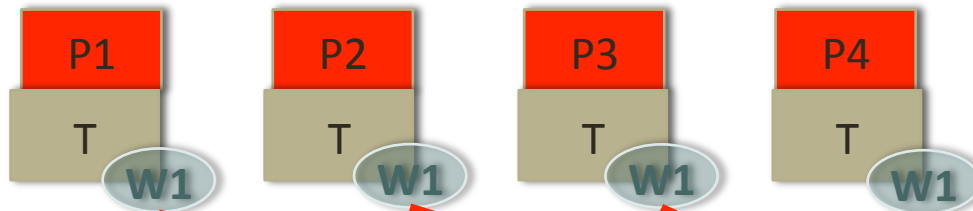
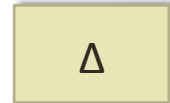
T

Δ



Concrètement qu'est-ce qui se passe quand on utilise DSMW ?

Quand Mr Jaune partage son wiki et que Mr Rouge récupère, il se passe..



Concrètement qu'est-ce qui se passe quand on utilise DSMW ?

Interface de négociation avec l'utilisateur

T

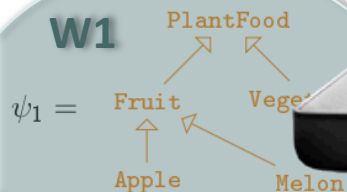
Δ

Classification des possibilités en fonction :

- Politiques de validité des tests
 - Prédéfinies
 - Choies par l'utilisateur
- Traces / expérience personnelle
- Traces / expérience des autres

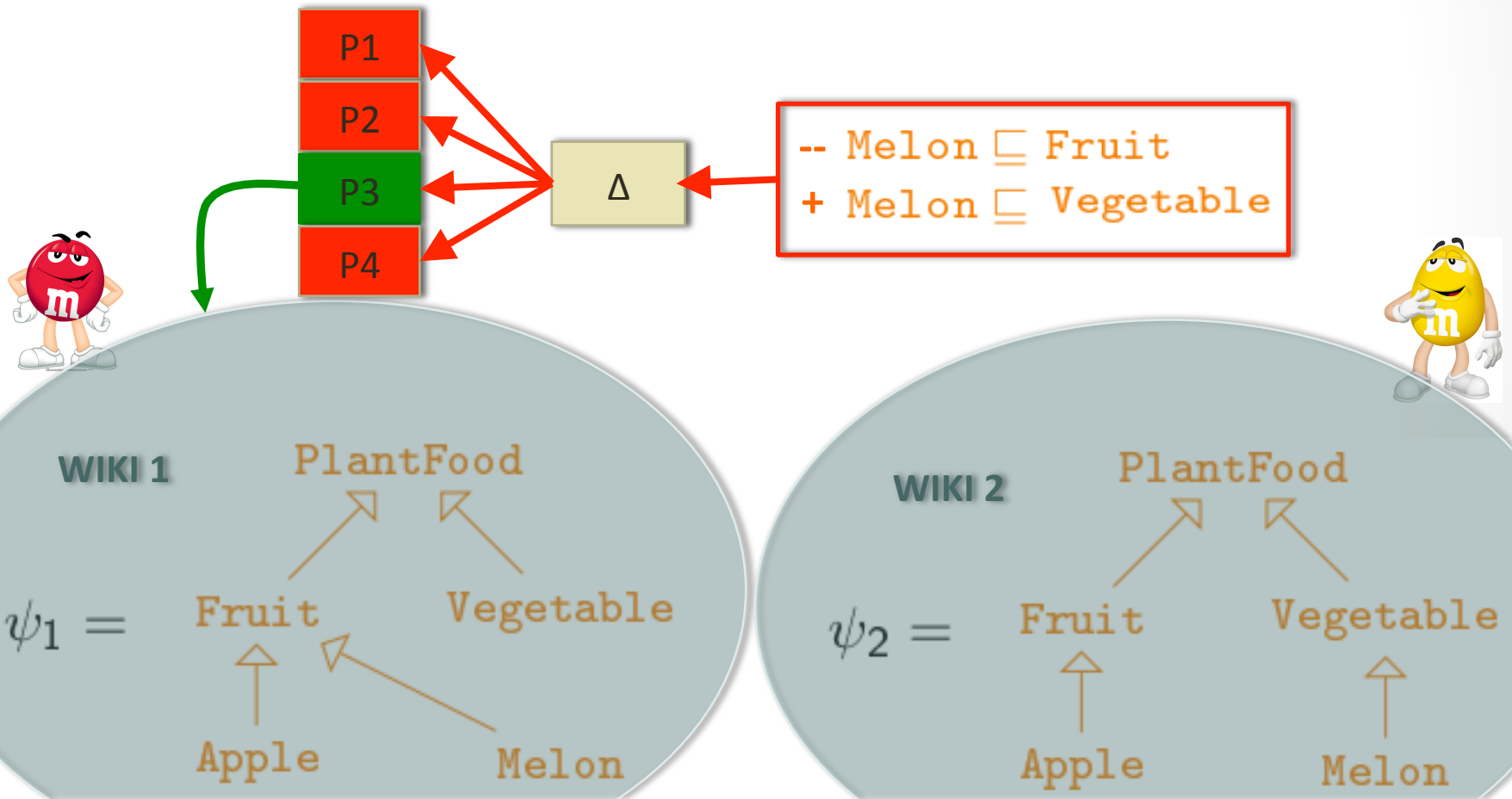


W1



Concrètement qu'est-ce qui se passe quand on utilise DSMW ?

La possibilité choisie par l'utilisateur est alors intégrée sous forme d'un patch



Et comment mettre tout ca en œuvre ?

- Coté détection de conflit
 - Julien arrête sa thèse...
... donc rien de codé avant Janvier ☹️
 - MAIS Formats d'entrée / sortie de la boîte noire fixés 😊
 - On peut donc faire comme si ca existait avec des fichiers fictifs...
- Coté K-CIP
 - Mise en œuvre prévue ?
 - Qui, quand, où, comment...
 - Si oui, fixer une formalisation des entrées/sorties ?

Quid d'un prototype commun ?

- WikiTaaable ?
- Ou autre ?
- Qui ?

Délivrables Silex T+18

- 4.4: Test and evaluation of the alter ego assistant with regard to the scenarios
 - Tests Collectra : stockage, interrogation, récupération, partage
 - => ca passe !
- 5.3: Algorithms and architecture for a distributed semantic wiki and a distributed alter-ego assistant
 - On a une architecture grossière
 - On pourra la définir complètement quand on aura fixé les types d'assistance...

Questions SILEX

à discuter à la plénière

- Est ce que le RDF Store de DSMW peut être substitué par un autre Store implémentant SparqlUpdate ? (par ex Corese)
- Cycle dans les taxonomies ?