

JSrealB

APPROCHE SYSTÉMATIQUE POUR LA RÉALISATION MULTILINGUE DE TEXTES

Étudiant : Paul Molins, INSA de Lyon, paul.molins@insa-lyon.fr

Directeur de recherche : Guy Lapalme, Université de Montréal, lapalme@iro.umontreal.ca

Présentation le 08/07/2015

Contexte

Projet de fin d'études au sein du laboratoire du RALI.

Projet reposant sur 2 projets distincts

- **MeteoVis** : un projet d'amélioration de la présentation de données météorologiques sur le Web, pour Environnement Canada, par Mohamed Mouine.
- **JSreal** : un réalisateur de texte en français pour la programmation Web, par Nicolas Daoust.

Objectif : Développer un réalisateur de textes bilingue inspiré de JSreal et l'intégrer à MeteoVis

METEOVIS

Thèse de PhD de Mohamed Mouine (juin 2014)



27.9 °C

Montréal, QC

Lundi 6 juillet 2015 à 16 h

°F en

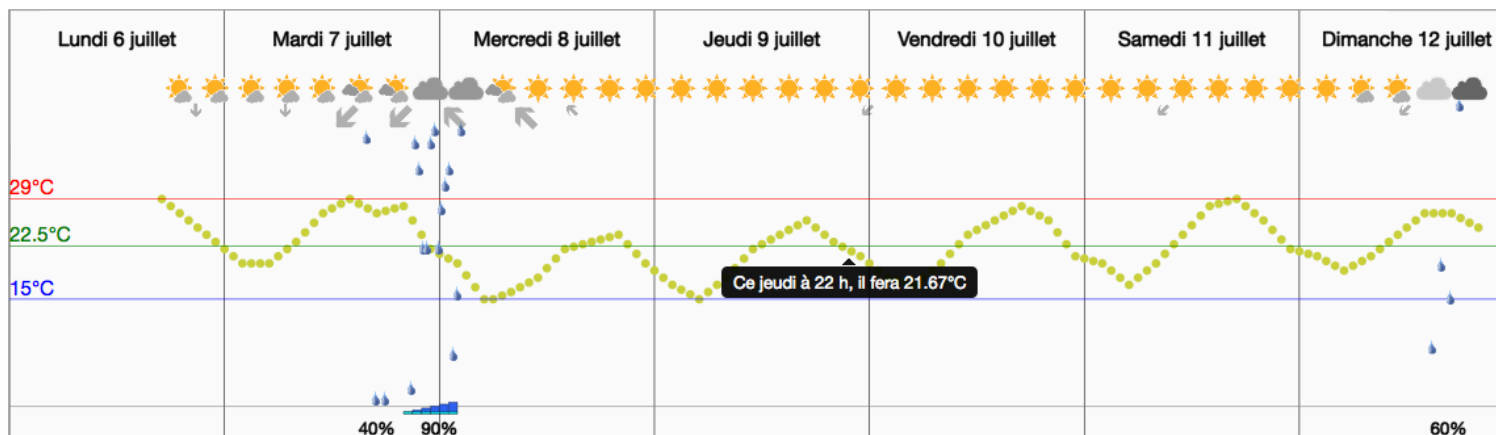
Condition	Généralement ensoleillé
Pression	101.8 kPa
Tendance	à la baisse
Visibilité	24.1 km

Température	27.9 °C
Point de rosée	15.8 °C
Humidité	47 %
Vent	SO 13 km/h

Pour les 3 prochaines heures	
17h	29.00
18h	28.00
19h	27.00

Changer de ville

7 jours

 Couverture nuageuse
 Précipitations
 Vent
 Température
 Accumulation
 % de précipitations


Jour	Température minimum	Température maximum	Informations complémentaires
Lundi 6 juillet			
Mardi 7 juillet			
Mercredi 8 juillet			
Jeudi 9 juillet			
Vendredi 10 juillet			
Samedi 11 juillet			
Dimanche 12 juillet			

Réingénierie

- ✓ Refactorisation du code
 - Programmation orientée objet
 - Architecture à 3 couches
 - Amélioration et optimisations diverses
- ✓ Mise à jour des données
 - Serveurs d'Environnement Canada surveillés en arrière plan
 - Téléchargement et stockage des prévisions sur la base de données *MeteoVis*
 - Téléchargement et stockage des données courantes sur le serveur *MeteoVis*
 - Temps de chargement passe de 5-30 secondes à 1,5 secondes
- ✓ Intégration de JSreal sur l'interface française de *MeteoVis*

JSrealB

Version bilingue de *JSreal*

Mémoire de maîtrise de Nicolas Daoust (sept 2013)

Présentation

- Réalisateur de textes
 - pour le Web
 - écrit en JavaScript
 - exécution par le navigateur
 - installation et utilisation simplifiées
- Solution bilingue (anglais/français)
- Basé sur la représentation syntaxique de JSreal

Représentation syntaxique

Français

```

S(
  CP(
    NP( AP( A("fort") ),
          N("averse") ),
    NP( N("orage") ),
    C("ou")
  ).n("p"),
  VP(
    V("cesser").t("pr"),
    PP( P("vers"),
        NP( N("minuit") )
      )
  )
).a("!")

```

Fortes averses ou orages cessant vers
minuit !

Représentation syntaxique

Anglais

```

S(
  CP(
    NP( AP( A("heavy") ),
          N("shower") ),
    NP( N("thunder") ),
    C("or")
  ).n("p"),
  VP(
    V("end").t("pr"),
    PP( P("near"),
        NP( N("midnight") )
      )
  )
).a("!")

```

Heavy showers or thunderstorms
ending near midnight!

Représentation syntaxique

Français

```

S(
  CP(
    NP( AP( A("fort") ),
          N("averse") ),
    NP( N("orage") ),
    C("ou")
  ).n("p"),
  VP(
    V("cesser").t("pr"),
    PP( P("vers"),
        NP( N("minuit") )
      )
  )
).a("!")

```

Fortes averses ou orages cessant vers
minuit !

Anglais

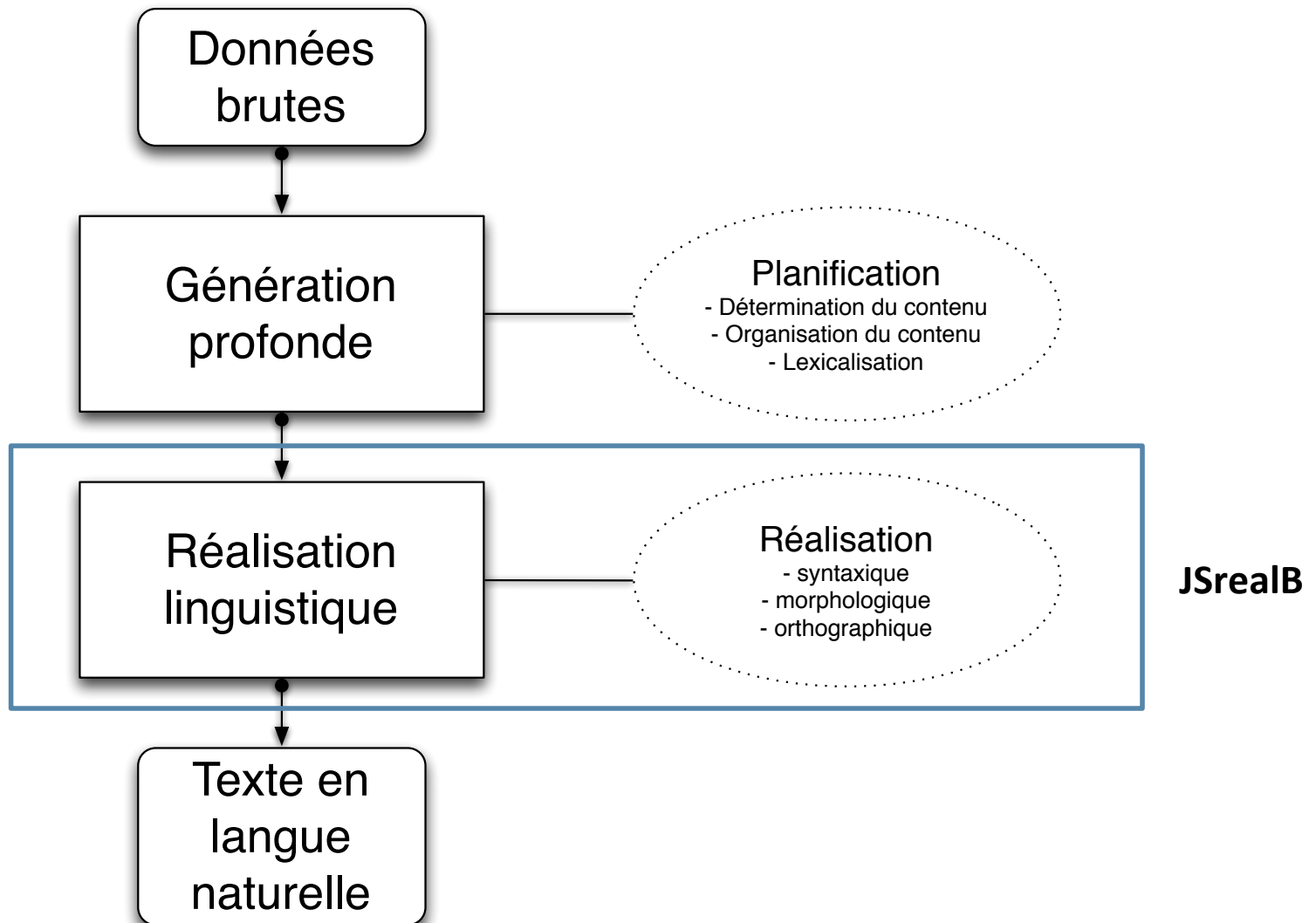
```

S(
  CP(
    NP( AP( A("heavy") ),
          N("shower") ),
    NP( N("thunder") ),
    C("or")
  ).n("p"),
  VP(
    V("end").t("pr"),
    PP( P("near"),
        NP( N("midnight") )
      )
  )
).a("!")

```

Heavy showers or thunderstorms
ending near midnight!

Génération automatique de textes

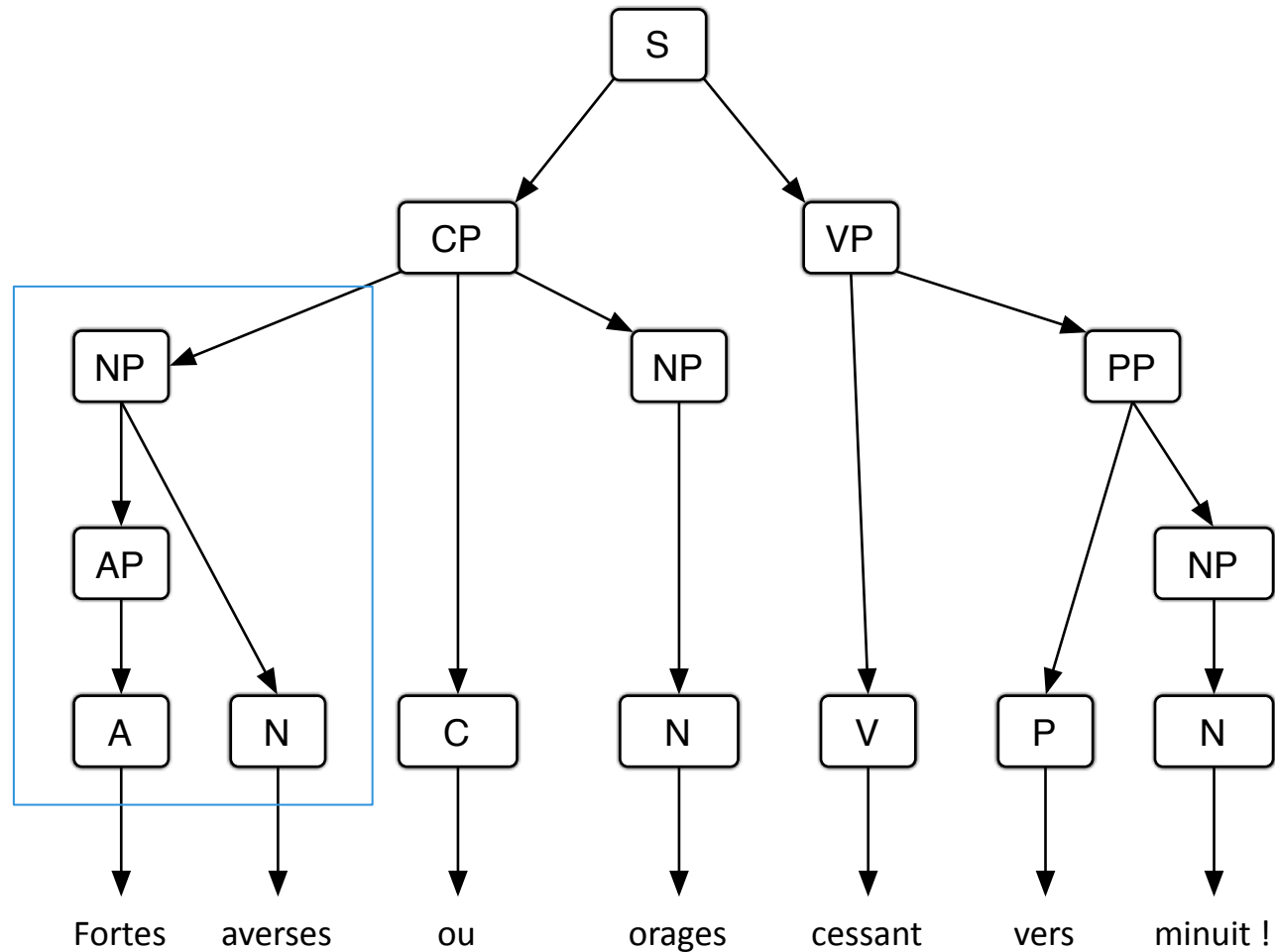


Réalisation syntaxique

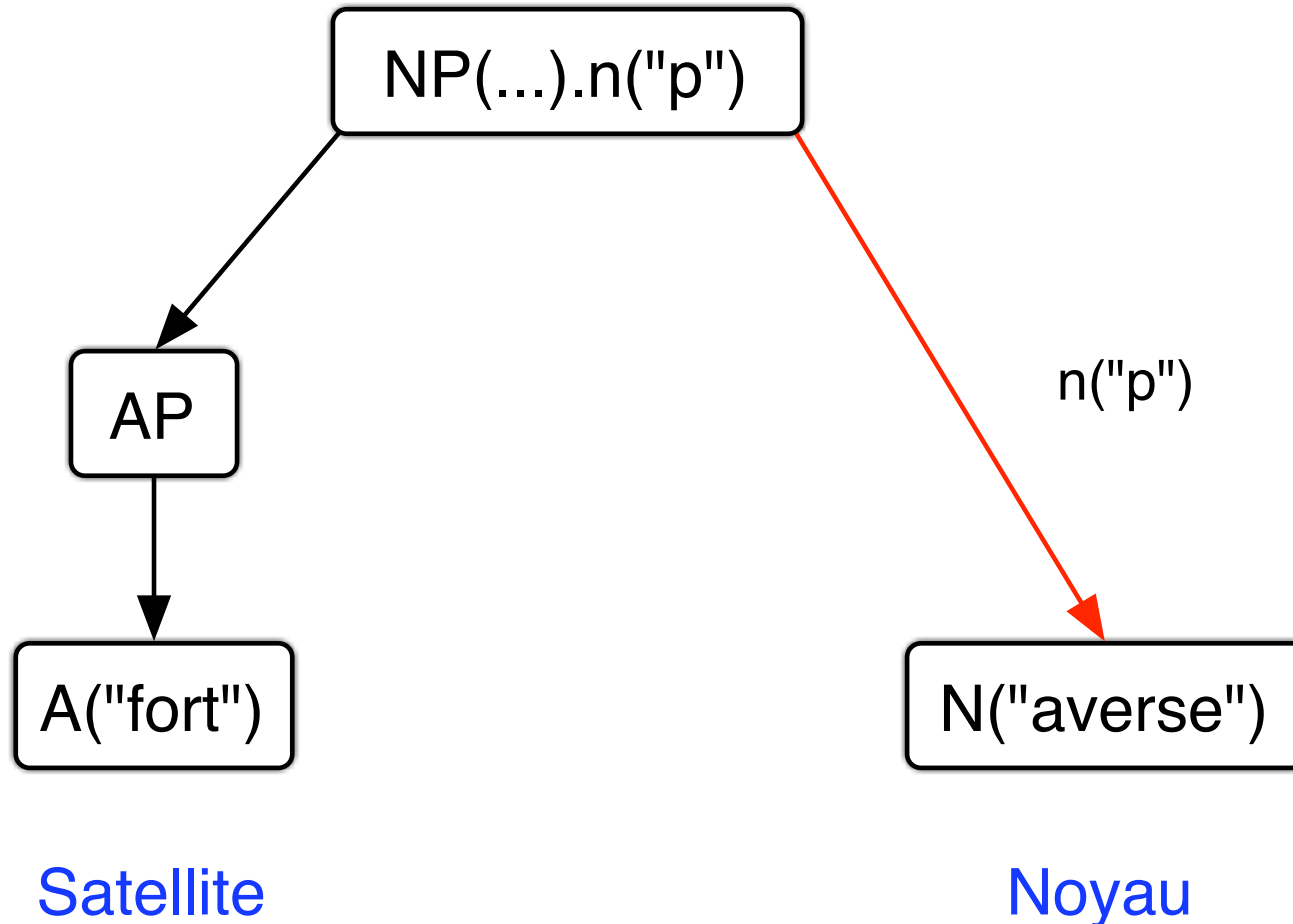
Rôles

1. Identification des catégories grammaticales
(indiquées par la représentation syntaxique)
2. Identification des fonctions grammaticales
3. Gestion de l'ordre des mots
(en partie donné par la représentation syntaxique)
4. Propagation des propriétés pour les accords entre les unités lexicales

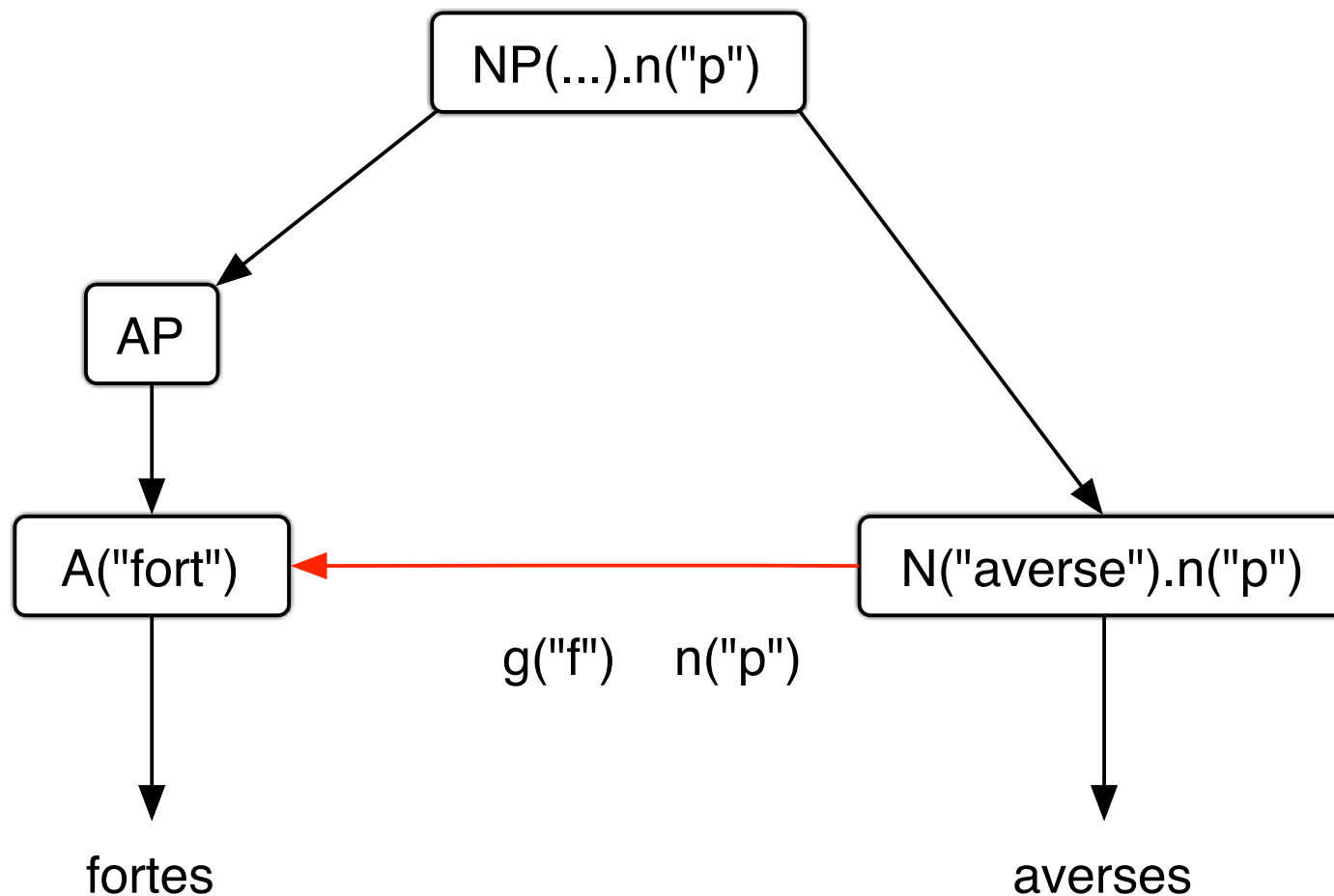
Réalisation syntaxique



Réalisation syntaxique



Réalisation morphologique



Réalisation morphologique

- Flexion du nom « joueur » au féminin pluriel

```
N("joueur").g("f").n("p")
```

```
joueuses
```

- Lexique

```
"joueur": { "N": { "g": "m", "tab": ["n55"] } }
```

- Table de règles

```
"n55": {
  "ending": "eur",
  "declension": [
    {"val": "eur", "g": "m", "n": "s"},
    {"val": "euse", "g": "f", "n": "s"},
    {"val": "eurs", "g": "m", "n": "p"},
    {"val": "euses", "g": "f", "n": "p"} ]
}
```

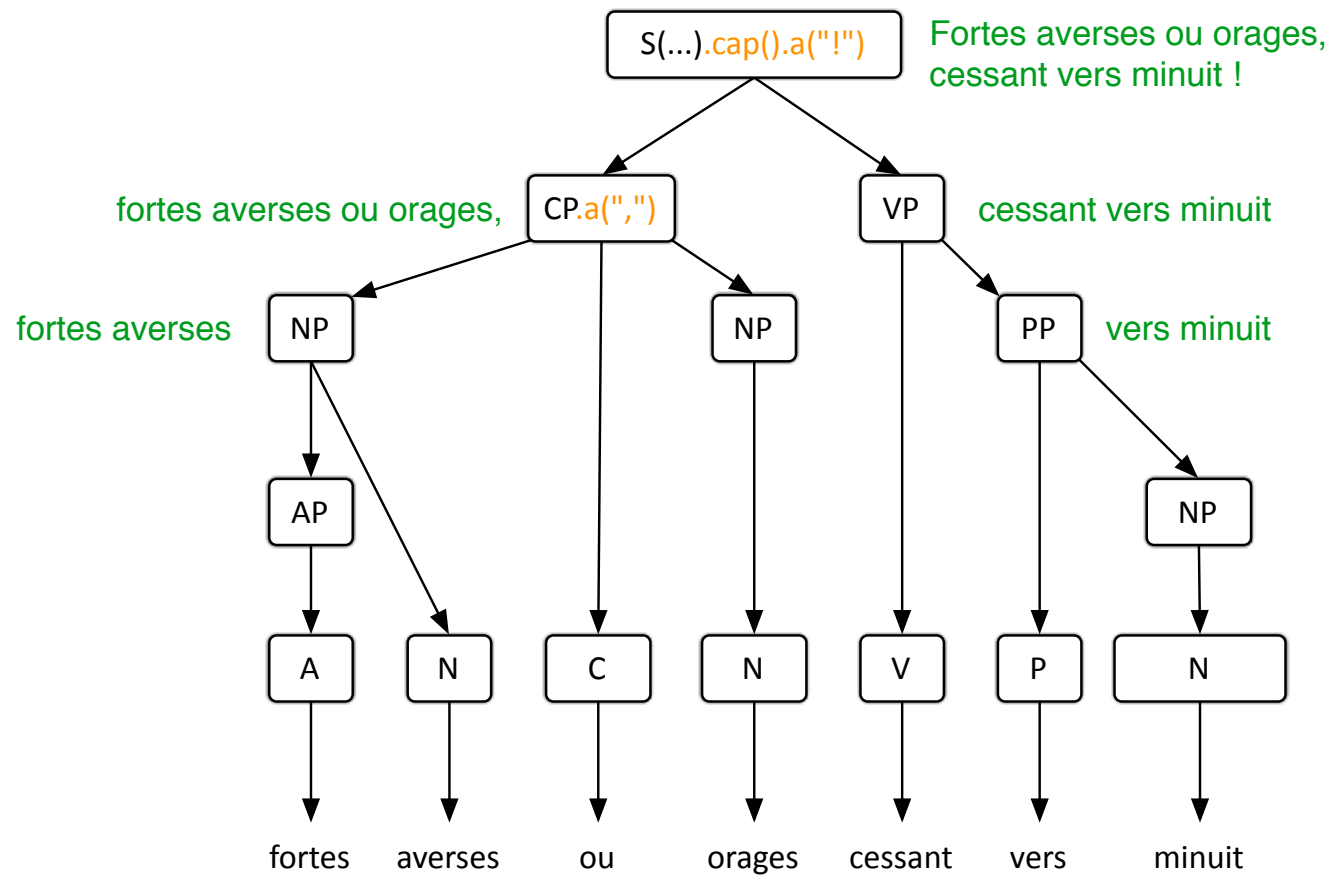

Réalisation morphologique

Les règles des tables prévalent sur les règles d'accord.

- Si le module de déclinaison reçoit en entrée un élément invariable (ex. l'adjectif *orange*) pour le décliner au pluriel, il retournera *orange*
- Si le module de déclinaison reçoit le nom *eau* pour une déclinaison au masculin, une erreur sera levée, car d'après les tables de règles, cette déclinaison n'existe pas

Réalisation orthographique

- Les caractéristiques orthographiques sont indiquées par l'utilisateur ou déterminées par le système (p. ex. pour la phrase)

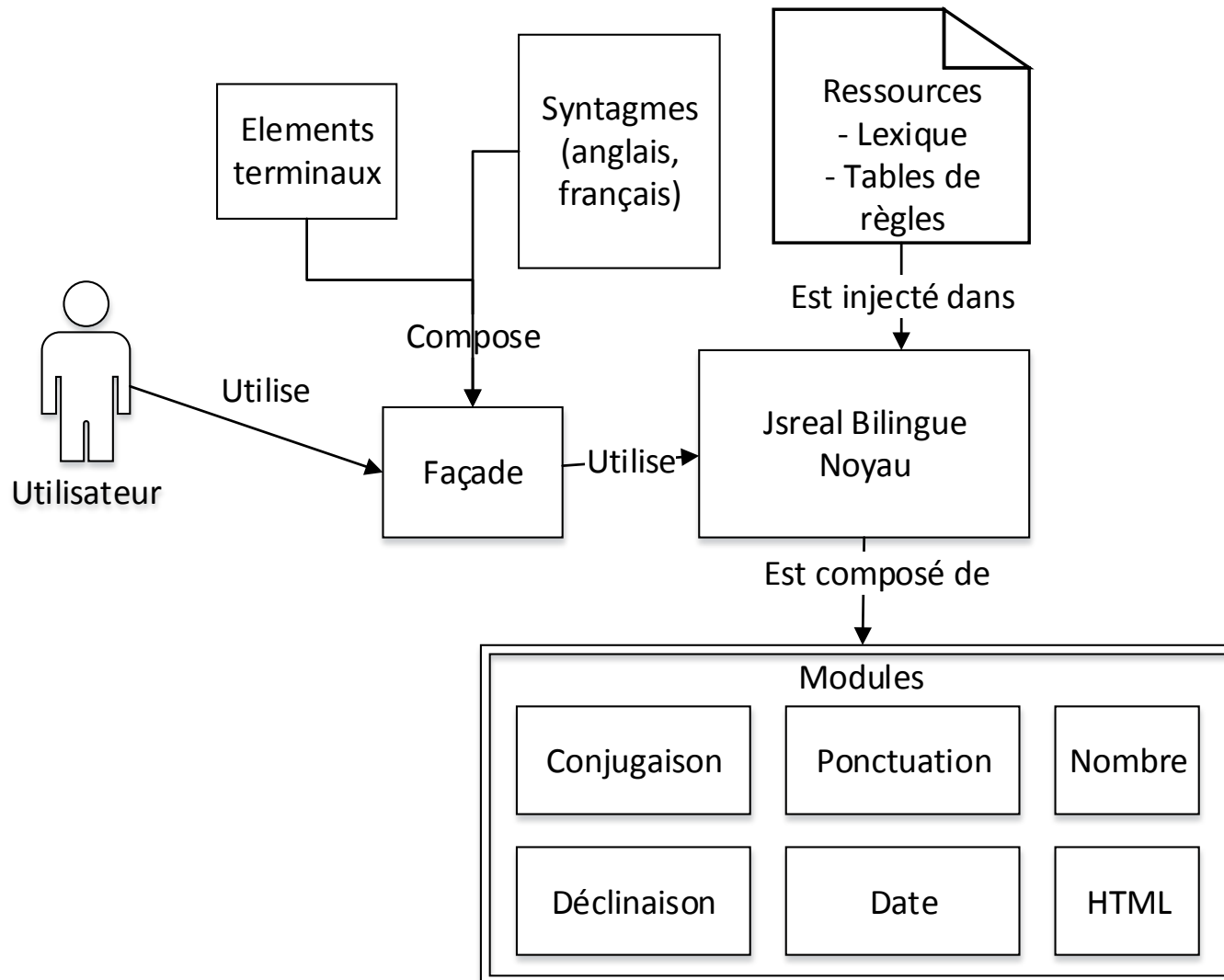


Formation des lexiques et tables

Objectifs

- Normaliser les ressources pour factoriser les algorithmes
- Réduire la taille pour accélérer le téléchargement
- Lexique : nomenclature de SimpleNLG-EnFr
- Tables de règles formés à partir de DME et DMF du RALI

Architecture



Couverture

La réalisation se fait à la volée en anglais ou en français

- Couverture complète de la représentation syntaxique
 - Toutes les catégories grammaticales
 - Tous les syntagmes
 - La phrase pour unifier les syntagmes
- Flexions
 - Conjugaison aux temps simples
 - Déclinaison en genre, nombre, au comparatif et superlatif
- Modules *Date* et *Number*
- Mise en forme
 - Ponctuation
 - Casse

Limites

Points non supportés

- Temps composés
- Phrases à plusieurs propositions
- Négation
- Forme interrogative
- ...

Objectif : Etendre la couverture des 2 langues pour atteindre celle de SimpleNLG-EnFr

Evaluation

JSrealB a été implémenté en suivant la technique du développement piloté par les tests.

Les tests ont été définis avant l'implémentation de JSrealB.

Pourquoi ?	Comment ?
<u>Contrôle</u> du fonctionnement dans un cas d'utilisation réel	Intégration dans MeteoVis
<u>Adéquation</u> de JSrealB avec les spécifications	<u>L'interface utilisateur</u> est testée par des <u>tests fonctionnels</u> (réalisation de syntagmes et phrases)
<u>Fonctionnement conjoint</u> d'un ensemble de composants	Chaque <u>module</u> est testé par des <u>tests d'intégration</u> (application de propriétés sur des unités lexicales)
<u>Fonctionnement individuel</u> de chaque composant	Chaque <u>fonction</u> de base est testée à l'aide de <u>tests unitaires</u>

Conclusion

JSrealB propose une approche en partie générique pour la réalisation bilingue de textes.

Intégration de JSrealB dans MeteoVis en cours.

Cette approche devrait permettre la réalisation dans toutes les langues flexionnelles.

Références

- [1] M. Mouine, *Présentation personnalisée des informations environnementales, Informatique et recherche opérationnelle*, Université de Montréal, 2014, pp. 129.
- [2] N. Daoust and G. Lapalme, *JSreal: A Text Realizer for Web Programming, Language Production, Cognition, and the Lexicon*, Springer, 2014, pp. 363-378.
- [3] E. Reiter and R. Dale, *Building natural language generation systems*, Cambridge University Press, 2000.
- [4] A. Gatt and E. Reiter, *SimpleNLG: A realisation engine for practical applications*, in A. f. C. Linguistics, ed., *12th European Workshop on Natural Language Generation*, Athens, Greece, 2009, pp. 90-93.
- [5] P.-L. Vaudry and G. Lapalme, *Adapting SimpleNLG for bilingual English - French realisation*, in A. f. C. Linguistics, ed., *14th European Workshop on Natural Language Generation*, Sofia, Bulgaria, 2013, pp. 183-187.

Merci de votre attention

Avez-vous des questions ?

Annexe - Lexique

```
"joie": {  
  "N": {  
    "g": "f",  
    "tab": ["n17"]  
  }  
},
```

- Propriétés sur les mots du lexique

```
"back": {  
  "Adv": { "tab": ["b1"] },  
  "N": { "tab": ["n1"] },  
  "V": { "tab": "v1" }  
},
```

- 1 entrée, plusieurs catégories grammaticales

Annexe - Tables de règles

```
"a6": {
  "ending": "d",
  "declension": [ { "val": "d" },
                  { "val": "dder", "f": "co" },
                  { "val": "ddest", "f": "su" } ]
},
```

- Superlatif et comparatif de l'adjectif *red* en anglais

```
"ave": {
  "ending": "e »,
  "option": [ { "val": "e" },
              { "val": "" } ]
},
```

- Toutes les transformations peuvent être encodées

Dans JSrealB, pas de gestion au cas par cas selon la langue, toutes les règles sont encodées dans les tables

Annexe – Démonstration de JSrealB

