

MTK (Mexico ToolKit) : un package pour unifier les démarches d'exploration numérique de modèles

Juhui WANG*, Robert FAIVRE, Hervé MONOD*, Hervé Richard
*INRA, Unité MaIAGE-Jouy en Josas

Plan

- **Contexte de travail**

- *Contexte et méthodes de travail.*
- *Objectifs : facilité, généricité, extensibilité et interopérabilité.*

- **Le package « mtk »**

- *Architecture orientée-objet et ouverte au Web Computing*
- *Fonctionnalités : utilisation interactive, intégration avec des plate-formes de simulation, et intégration des contributions tierces.*

- **Exemples d'utilisation**

- *Modèle natif.*
- *Modèle programmé en R.*
- *Intégration avec des plate-formes de modélisation.*
- *Extension du mtk par l'intégration des méthodes développées par des tierces personnes.*

Contexte du travail

- **Méthodes et logiciels : abondants, divers et variés**

Sensitivity (G. Pujol *et al.*), Planor (H. Monod *et al.*), PLMM (R. Faivre *et al.*), SimLab (A. Saltelli *et al.*), OpenTurns (EDF R&D *et al.*), LHS (R. Carnell), DiceDesign (J. Franco *et al.*), Promethee (Y. Richet *et al.*)

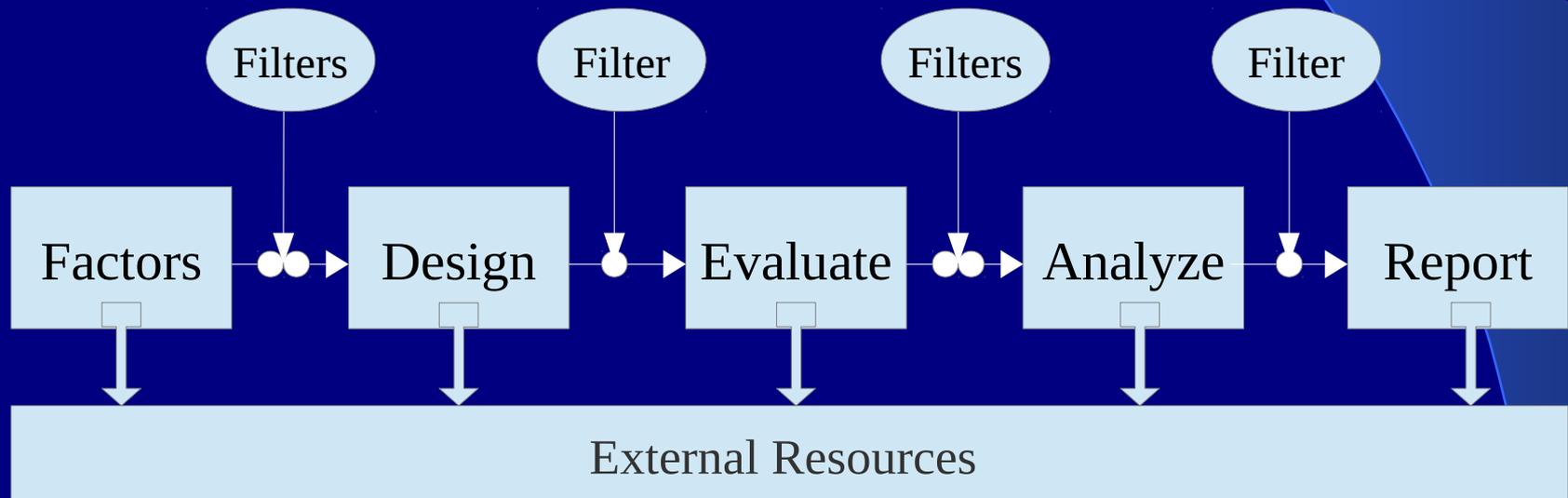
- **Difficultés :**

- Démarches différentes, Syntaxe d'utilisation différente.
- Difficulté d'utilisation, Difficulté de comparaison, Difficulté de choix

- **But :** une plate-forme générique et extensible pour unifier les méthodes existantes ou à venir

Méthode de travail

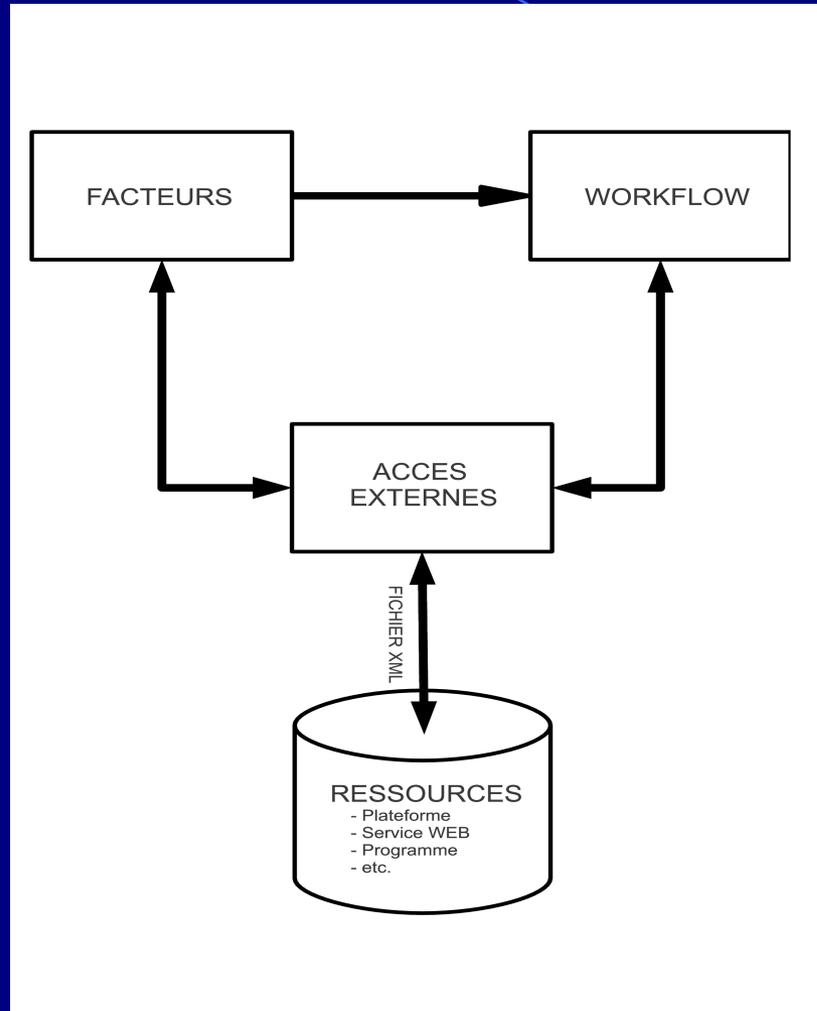
- Identifier les processus élémentaires.
- Implémenter les méthodes par combiner les processus élémentaires.
- Conduire les traitements au moyen d'un workflow.



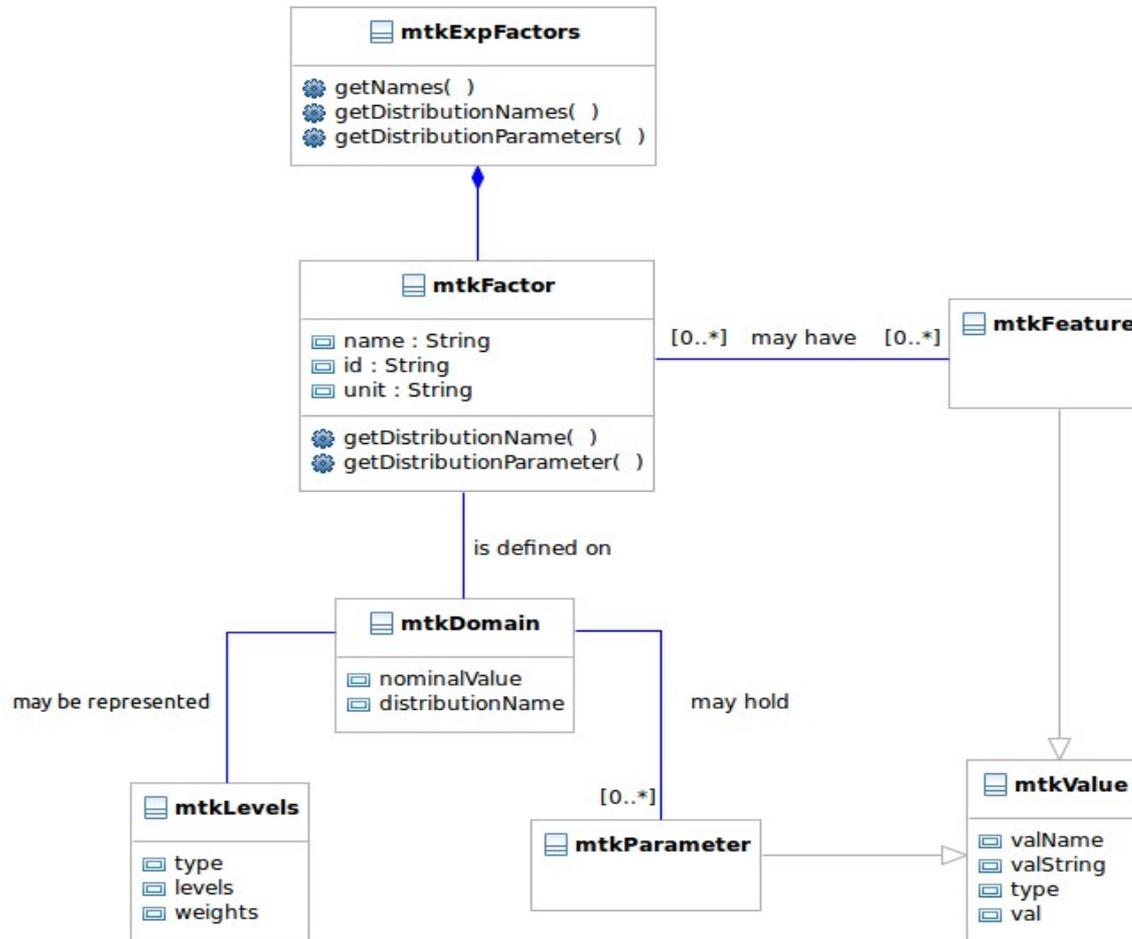
Objectifs

- **Facilité :**
 - Syntaxe d'utilisation unifiée.
 - Démarche d'analyse standardisée.
- **Généricité :**
 - Intégration des méthodes existantes .
- **Extensibilité :**
 - Gestion des facteurs de complexité « quelconque »
 - Intégration des nouvelles méthodes (réalisées en R)
- **Interopérabilité :**
 - Intégration transparente avec les plate-formes existantes (plate-forme de modélisation)
- **Efficacité de calcul par HPC**
 - *Multi-core, cluster, grid.*
 - *Embarquement non intrusif des modèles.*

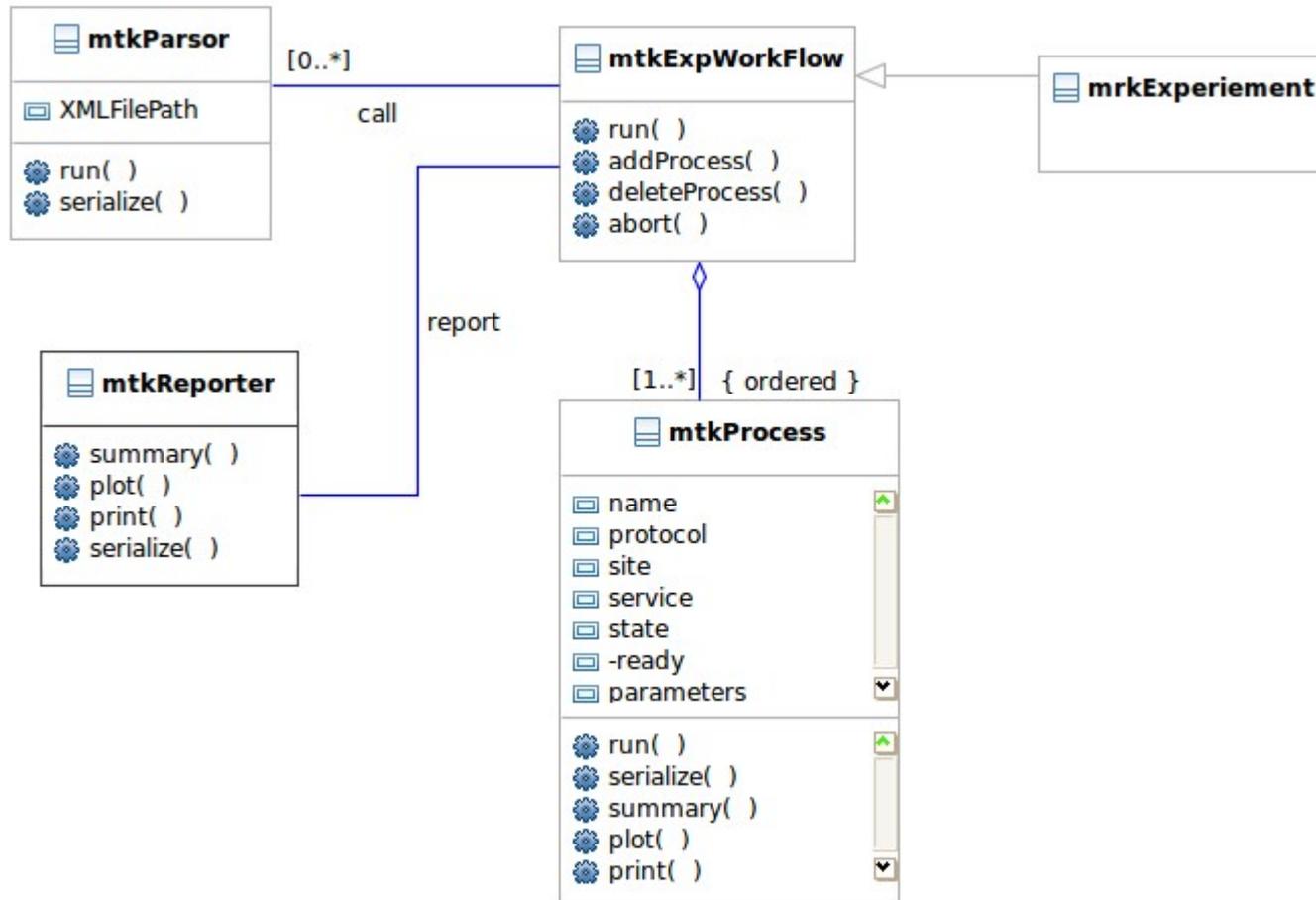
Architecture « mtk »



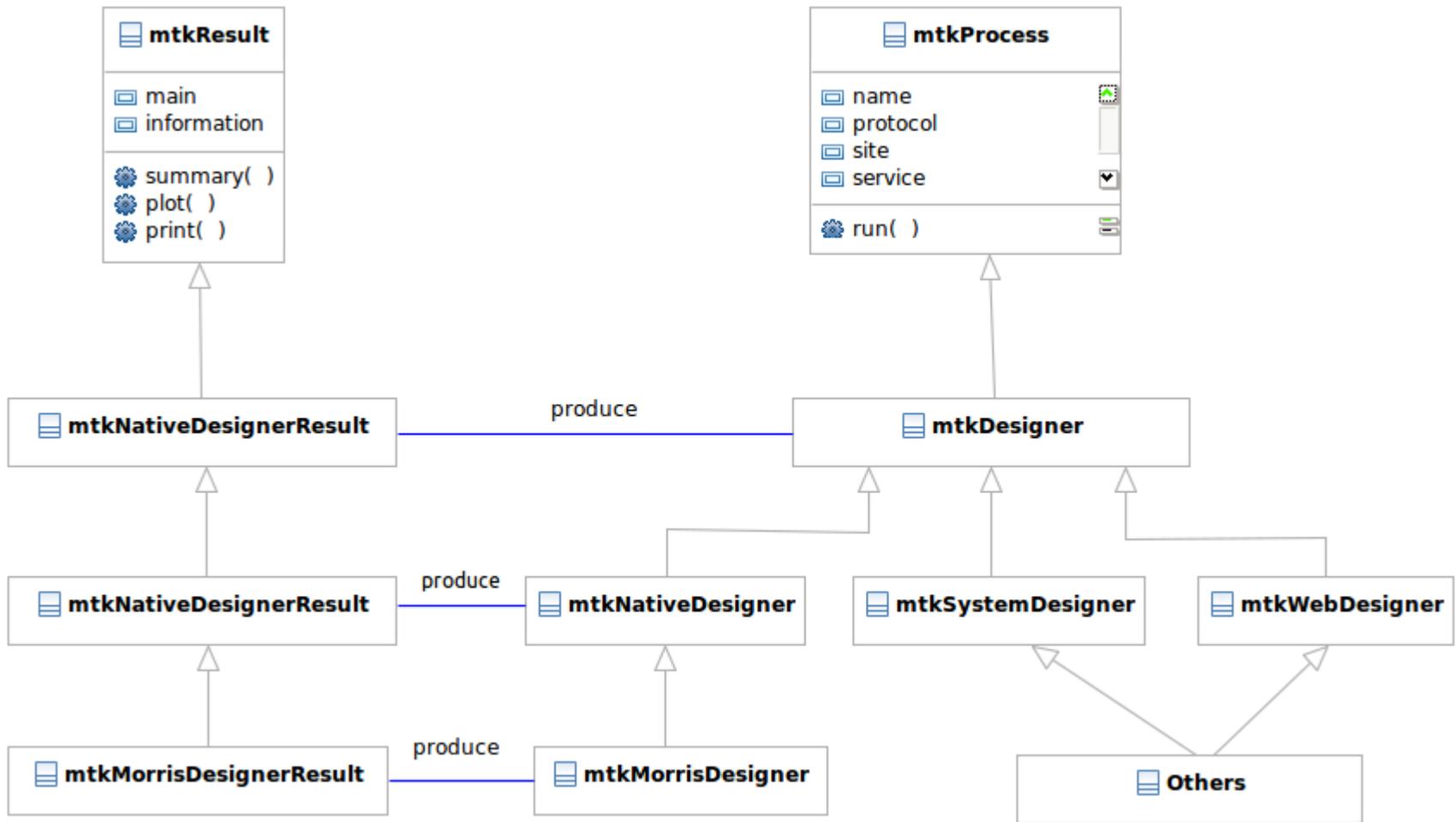
Conception (1)



Conception (2)



Conception (3)



Fonctionnalités

- Gestion de facteurs complexes
- Gestion générique des traitements par workflow
- Gestion des modèles complexes et simples
- Reporting des résultats
- Intégration avec des ressources existantes
- Extension avec des contributions des tiers
- Prise en charge des données XML

- *Prise en charge du protocole Web Services*
- *Distribution du calcul sur une structure HPC*
- *Embarquement non-intrusif des modèles dans un environnement HPC*

Logique d'utilisation du « mtk »

- **Présentation uniforme des méthodes et des données**

- *La méthode "XXX" pour la génération du plan d'expérience:*
 - *mtkXXXDesigner(listParameters=list())*
 - *mtkXXXDesignerResult()*
- *La méthode "YYY" pour la simulation*
 - *mtkYYEvaluator(listParameters=list())*
 - *mtkYYEvaluatorResult()*
- *La méthode "ZZZ" pour l'analyse de sensibilité*
 - *mtkZZZAnalyser(listParameters=list())*
 - *mtkZZZAnalyserResult()*

- **Démarche en 4 étapes**

- *Spécifier les facteurs*
- *Spécifier les processus de traitement*
- *Rassembler les processus au sein d'un workflow*
- *Lancer les traitements et générer les rapports*

Exemple « Ishigami »

$$f(\mathbf{x}) = \sin(x_1) + a \sin^2(x_2) + b x_3^4 \sin(x_1)$$

```
ishigami <- function(xx, a=7, b=0.1)
{
  x1 <- xx[1]
  x2 <- xx[2]
  x3 <- xx[3]

  term1 <- sin(x1)
  term2 <- a * (sin(x2))^2
  term3 <- b * x3^4 * sin(x1)

  y <- term1 + term2 + term3
  return(y)
}
```

Exemple « Ishigami »

- **Spécifier les facteurs pour le modèle Ishigami**

```
x1 <- make.mtkFactor(name="x1", distribName="unif",  
                    distribPara=list(min=-pi, max=pi))  
x2 <- make.mtkFactor(name="x2", distribName="unif",  
                    distribPara=list(min=-pi, max=pi))  
x3 <- make.mtkFactor(name="x3", distribName="unif",  
                    distribPara=list(min=-pi, max=pi))  
ishi.factors <- mtkExpFactors(list(x1,x2,x3))
```

- **Spécifier les processus impliqués**

```
c <- mtkBasicMonteCarloDesigner(  
    listParameters = list(size=20))  
s <- mtkIshigamiEvaluator()  
a <- mtkRegressionAnalyser(  
    listParameters = list(nboot=20) )
```

Exemple « Ishigami »

- **Construire un workflow**

```
exp <- mtkExpWorkflow( expFactors = ishi.factors,  
                      processesVector = c( design=c,  
                                           evaluate=s, analyze=a))  
  
run(exp)
```

- **Reporter les résultats**

```
summary(exp)
```

Exemples

- **Modèle natif implémenté dans le package « mtk »**
- **Modèle programmé comme une fonction R**
- **Pilotage du « mtk » au moyen des fichiers XML**
- **Modèle comme une application système**
- **Importation des données générées off-line**
- **Intégration avec des plate-formes de modélisation**
- **Extension du package avec des méthodes développées par des personnes tierces (R)**

Extension du package « mtk »

- Le package « mtk » dispose des outils qui permettent de transformer des nouvelles méthodes programmées par des tiers en des classes « mtk compliant ».
 - `mtk.designerAddons()`
 - `mtk.evaluatorAddons()`
 - `mtk.analyserAddons()`
- Exemple d'ajout des méthodes « Experiment Design »
 - `mtk.designerAddons(where="samplerSobol.R",
authors="H. Monod", name="HMSobol",
main="sampler.sobol", plot="plot.sobol",
summary="summary.sobol")`
 - `Sampleur <- mtkHMSobolDesigner(listParameters =
list(N=200, shrink=0.8))`

Extension du package « mtk » avec un modèle externe

- `mtk.evaluatorAddons (where="WeedModel_v2.R",
authors="D. Makowski (2012)", name="Weed",
main="weed.simule")`
- `weed.exp <- mtkWeedEvaluator(
listParameters = list(decision=1))`

Conclusions et Perspectives

- **« mtk » fait partie des outils pour l'exploration numérique des modèles**
 - Centré sur la planification d'expérience, la simulation numérique, et l'analyse de sensibilité.
 - Réalisé selon la norme S4
 - Générique et extensible
 - Intègre les technologies de Web Computing et HPC
- **Travail à faire**
 - Gestion du HPC
 - Embarquement des modèles de façon non-intrusive dans un environnement HPC
 - Interface Web Services

Remerciement

- **Membres principaux :**

- **Juhui WANG, Hervé MONOD, INRA - Jouy en Josas**
- **Robert FAIVRE, INRA – Toulouse**
- **Hervé RICHARD, INRA – Avignon**

- **Contributeurs :**

**IRSTEA, IFRMER, CIRAD, INRA,
UNIVERSITÉ DU LITTORAL, CODE LUTIN.**