



## le package MTK et l'analyse de sensibilité

Juhui Wang, Herve Monod, Robert Faivre

### ► To cite this version:

Juhui Wang, Herve Monod, Robert Faivre. le package MTK et l'analyse de sensibilité. Rencontre avec CATI-IUMA du department EA, 2014, Paris, France. 16 diapos. hal-01259729

**HAL Id: hal-01259729**

**<https://hal.science/hal-01259729>**

Submitted on 5 Jun 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# MTK (Mexico ToolKit) : un package R pour la gestion de l'exploration numérique des modèles

Juhui WANG  
INRA, Unité MIA-Jouy en Josas

# Plan

- **Contexte de travail**

- *Fil conducteur*
- **Objectifs** : *facilité, généricité, extensibilité et interopérabilité.*

- **Le package « mtk »**

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

- **Exemples d'utilisation**

- *Modèle natif*
- *Modèle simple programmé en R*
- *Spécification du workflow via un fichier XML*
- *Modèle complexe programmé en R*
- *Extension par des addons*

# Contexte du travail

- **Méthodes existantes :**

- Abondantes
- Diversifiées

- **Difficultés :**

- Utilisation
- Comparaison

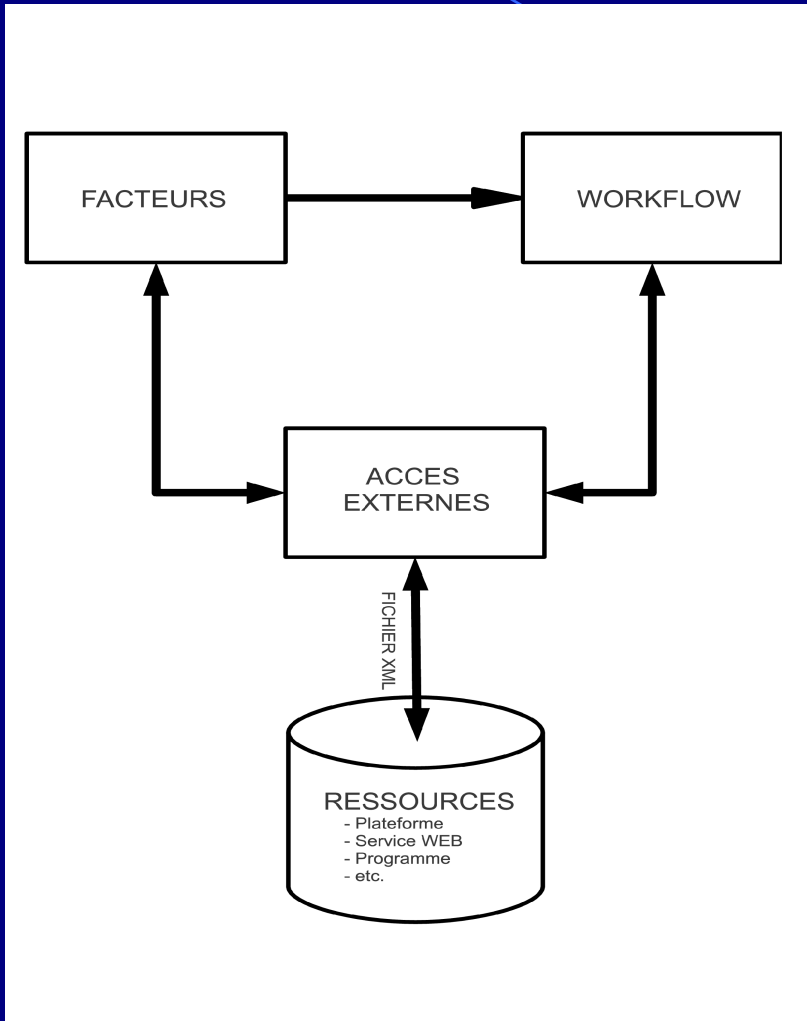
- **Structuration :**

- specify
- design
- evaluate
- analyze
- report

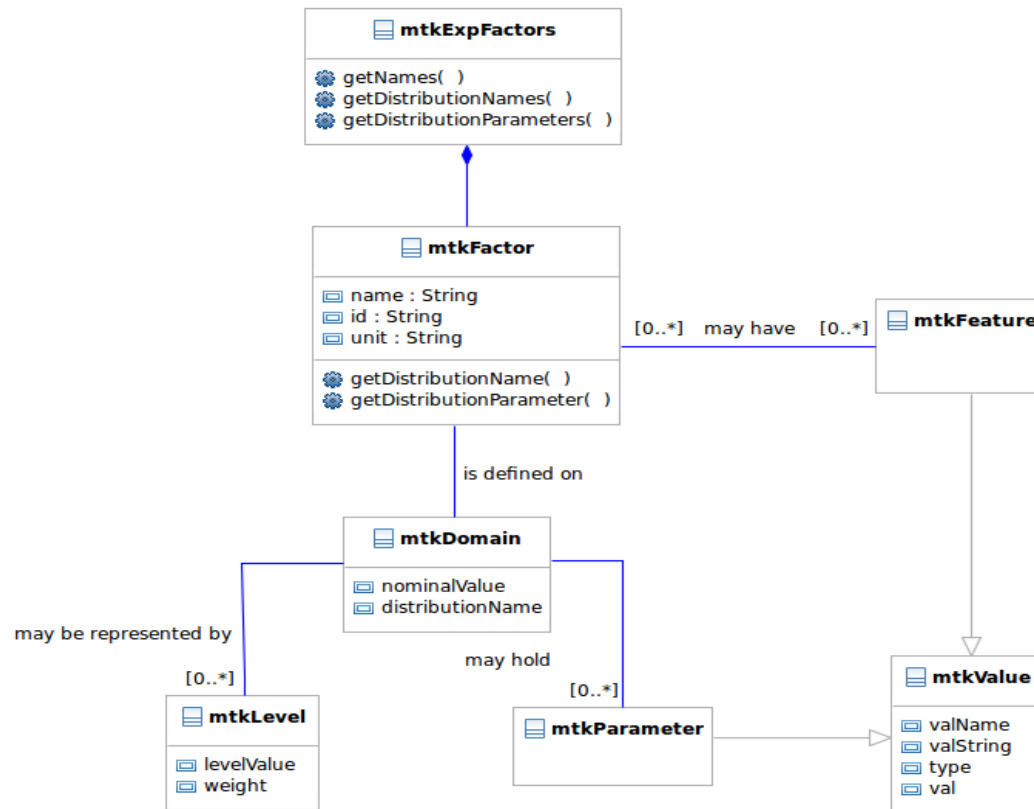
# Objectifs

- **Facilité :**
  - - Syntaxe d'utilisation unifiée.
  - - Structuration de la démarche : design, evaluate, analyze, report.
- **Généricité :**
  - Encapsulation des méthodes existantes.
- **Extensibilité :**
  - Intégration des nouvelles méthodes réalisées en R grâce aux outils indépendants disponibles au sein du package.
- **Interopérabilité :**
  - Intégration transparente avec les plates-formes existantes grâce à l'utilisation des fichiers XML

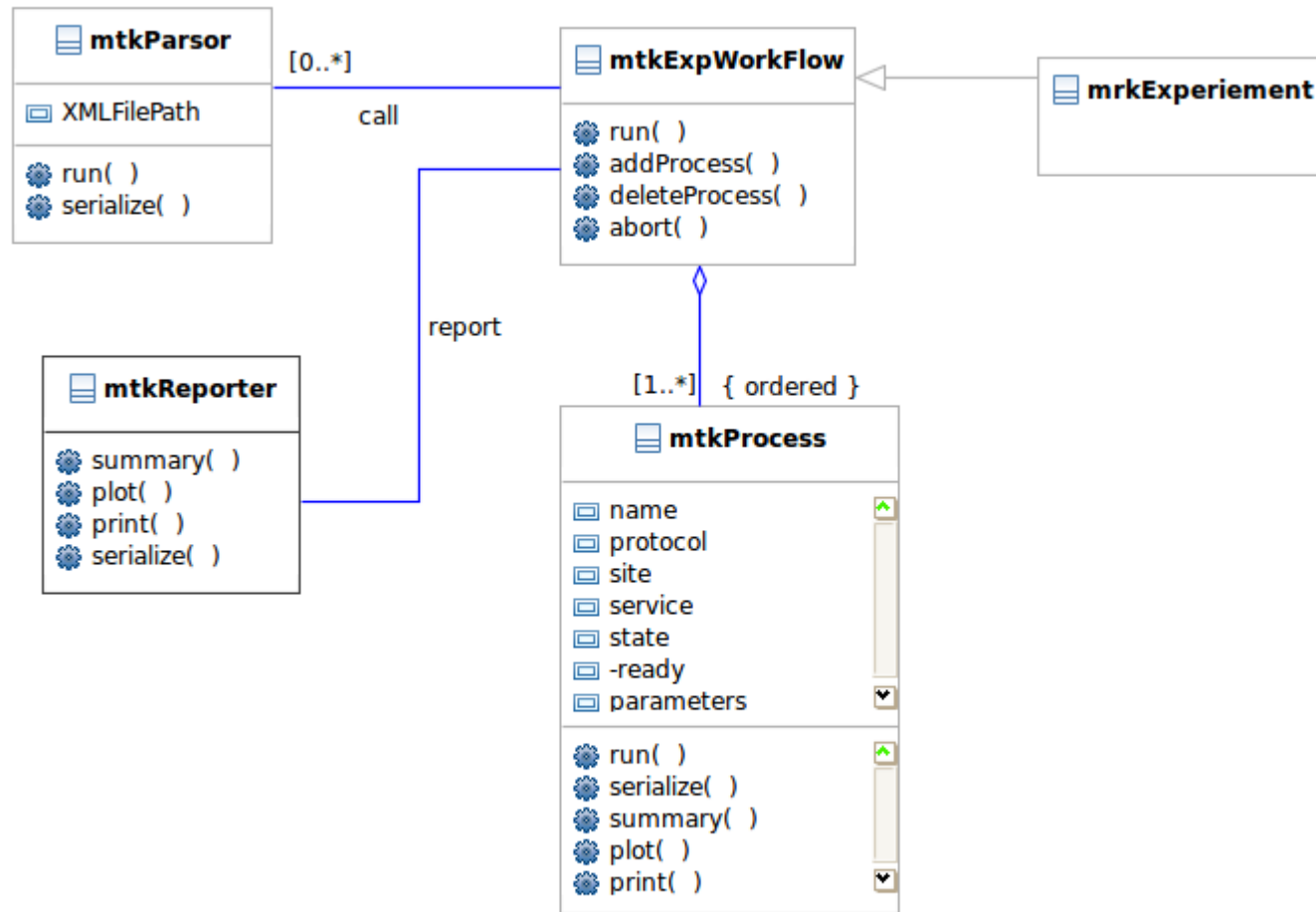
# Architecture « mtk »



# Conception (1)

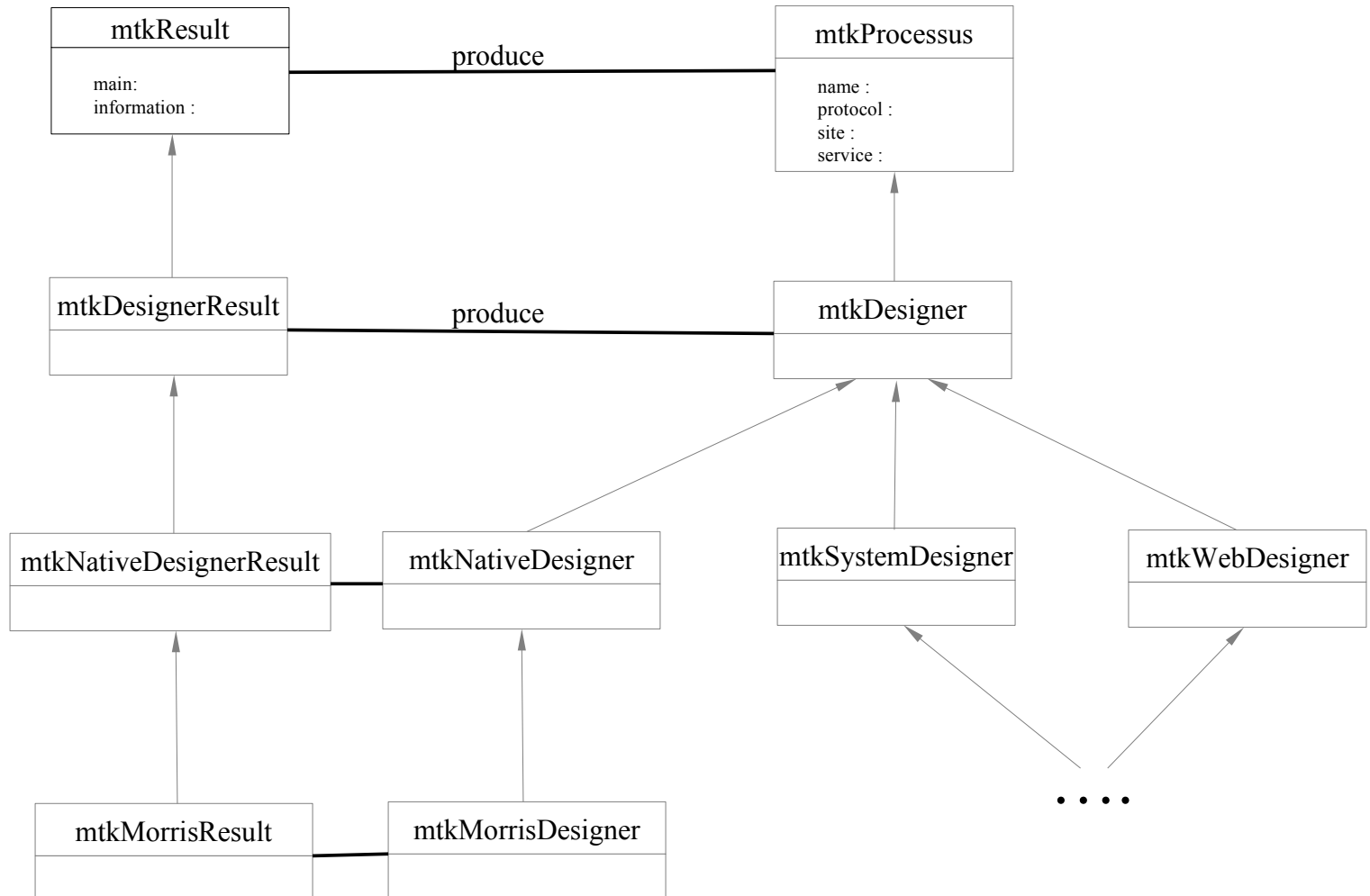


## Conception (2)





## Conception (3)



# Fonctionnalités

- **Gestion générique de la procédure d'analyse de sensibilité**
- **Présentation uniformisée des méthodes**
- **Mise en forme et présentation des résultats**
- **Intégration avec des ressources existantes**
- **Intégration des contributions des tiers.**

# La démarche mtk

- **Décomposition normalisée de processus de l'exploration numérique des modèles**

*Choix des facteurs -> Génération des plans -> Simulation de modèle ->  
--> Calcule de sensibilité -> Génération des rapports*

- **Formalisation des tâches, de leurs gestionnaires, et des résultats produits :**

– *design -----> designer -----> experiments design*  
– *evaluate -----> evaluator -----> model output*  
– *analyze -----> analyser -----> sensitivity index*

- **Présentation uniformisée des méthodes et des données**

*La méthode "XXX " pour la tâche "design " :*

- *mtkXXXDesigner(listParameters=list())*
- *mtkNativeDesigner("XXX", information=list())*
- *mtkXXXDesignerResult()*

- **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*

# Use-Cases

- **Modèle natif implémenté dans le package « mtk ».**
- **Modèle programmé comme fonction R indépendante**
- **Modèle dans une librairie ou programmés par des tiers**
- *Modèle comme une application système*
- *Modèle dans une plate-forme de modélisation*

- **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 qui réalisent les traitements**

```
concepteur <- mtkBasicMonteCarloDesigner(  
  listParameters = list(size=20))  
simulateur <- mtkIshigamiEvaluator()  
analyseur <- mtkRegressionAnalyser(  
  listParameters = list(nboot=20) )
```

- **Construire un workflow**

```
exp <- mtkExpWorkflow( expFactors = ishi.factors,  
  processesVector = c( design=concepteur,  
    evaluate=simulateur, analyze=analyseur)  
  )
```

- **Exécuter le workflow et reporter les résultats**

```
run(exp)  
summary(exp)
```

# Utilisation via un fichier XML

- Construire le workflow à partir du fichier XML

```
exp <- mtkExpWorkflow(xmlFilePath = "morris.xml")
```

- Exécuter le workflow et reporter les résultats
  - *run()*
  - *summary()*, *print()*, *plot()*, *show()*, *report()*

## Contributions au package « mtk »

- Le package « mtk » dispose des outils qui permettent de transformer des nouvelles méthodes programmées par des tiers en des classes compatibles avec le package « mtk ».
  - `mtk.designerAddons()`
  - `mtk.evaluatorAddons()`
  - `mtk.analyserAddons()`



## mtk.designerAddons()

```
mtk.designerAddons (where="Sobol.R",  
  authors="H. Monod, INRA-MIA", name="Sobol",  
  main="sobol", plot="pl.sobol",  
  summary="s.sobol")
```

```
sampleur ← mtkSobolDesigner(  
  listParameters = list(N=200, shrink=0.8)  
)
```