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

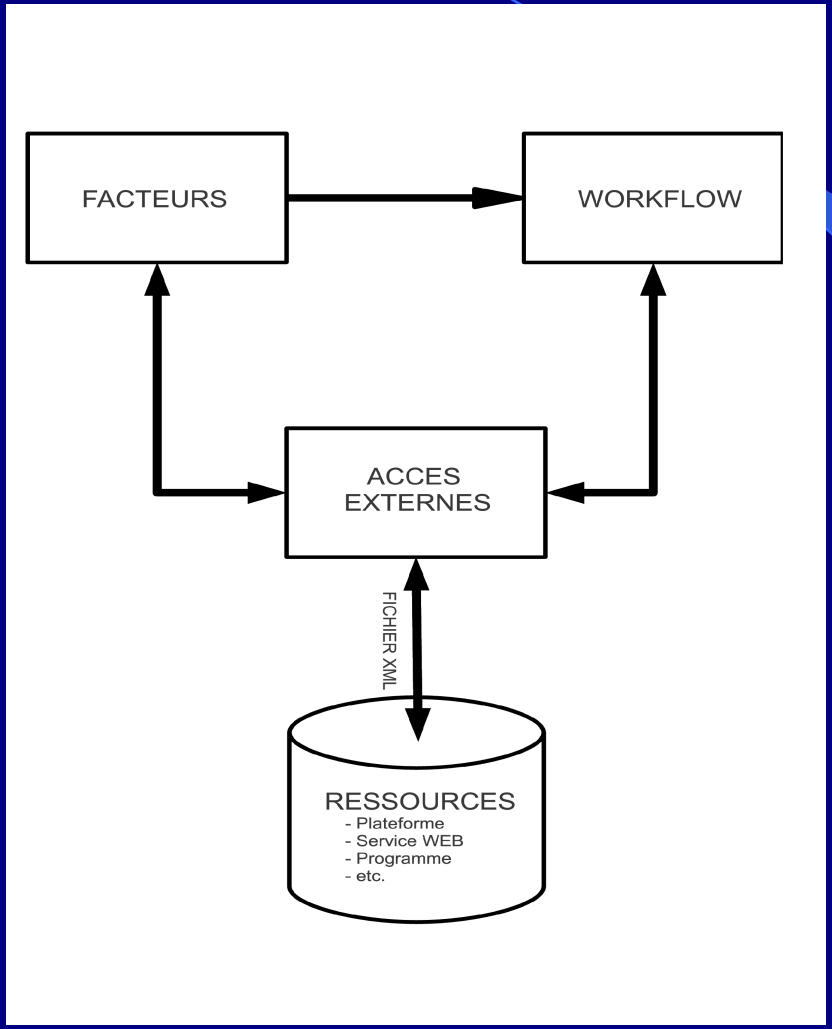
- **Facilité :**
  - - Syntaxe d'utilisation unifiée.
  - - Structuration de la démarche : design, evaluate, analyze, report.
- **Généricité :**

  - Encapsulation des méthodes existantes.
- **Extensibilité :**

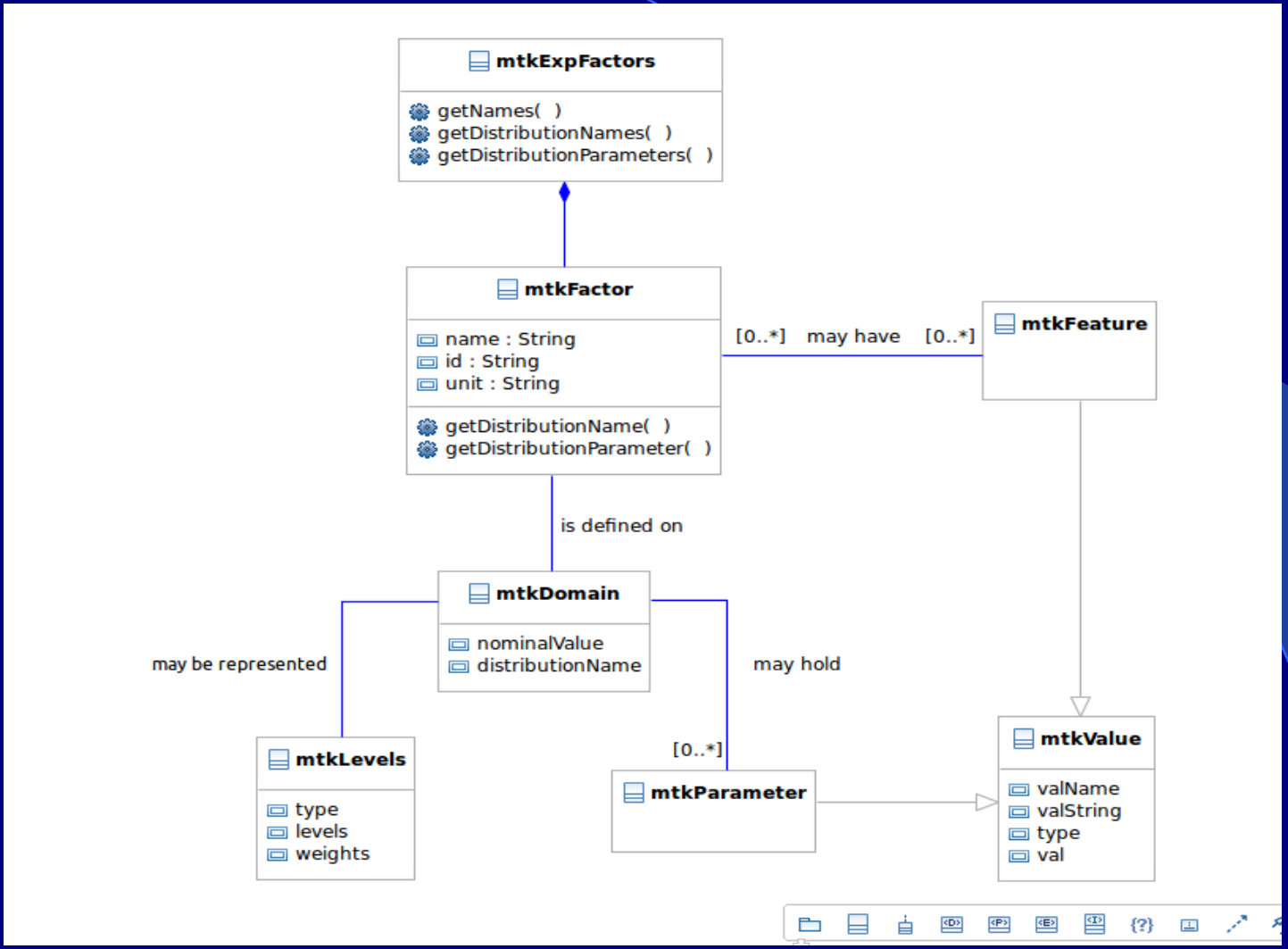
  - Intégration des nouvelles méthodes réalisées en R grâce aux outils indépendants disponibles au sein du package.
- **Interopérabilité :**

  - Intégration transparente avec les plates-formes existantes grâce à l'utilisation des fichiers XML

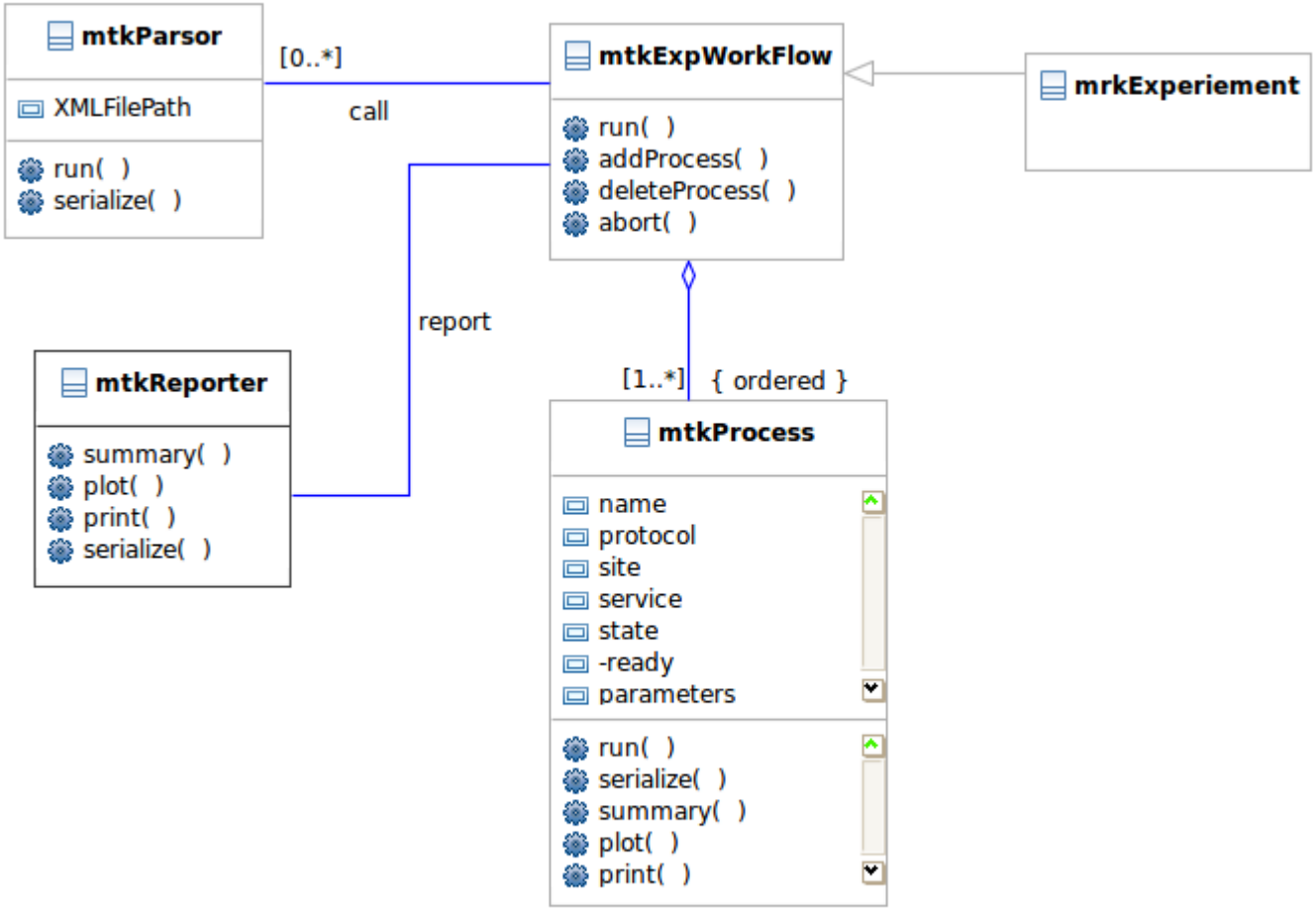
Architecture « mtk »



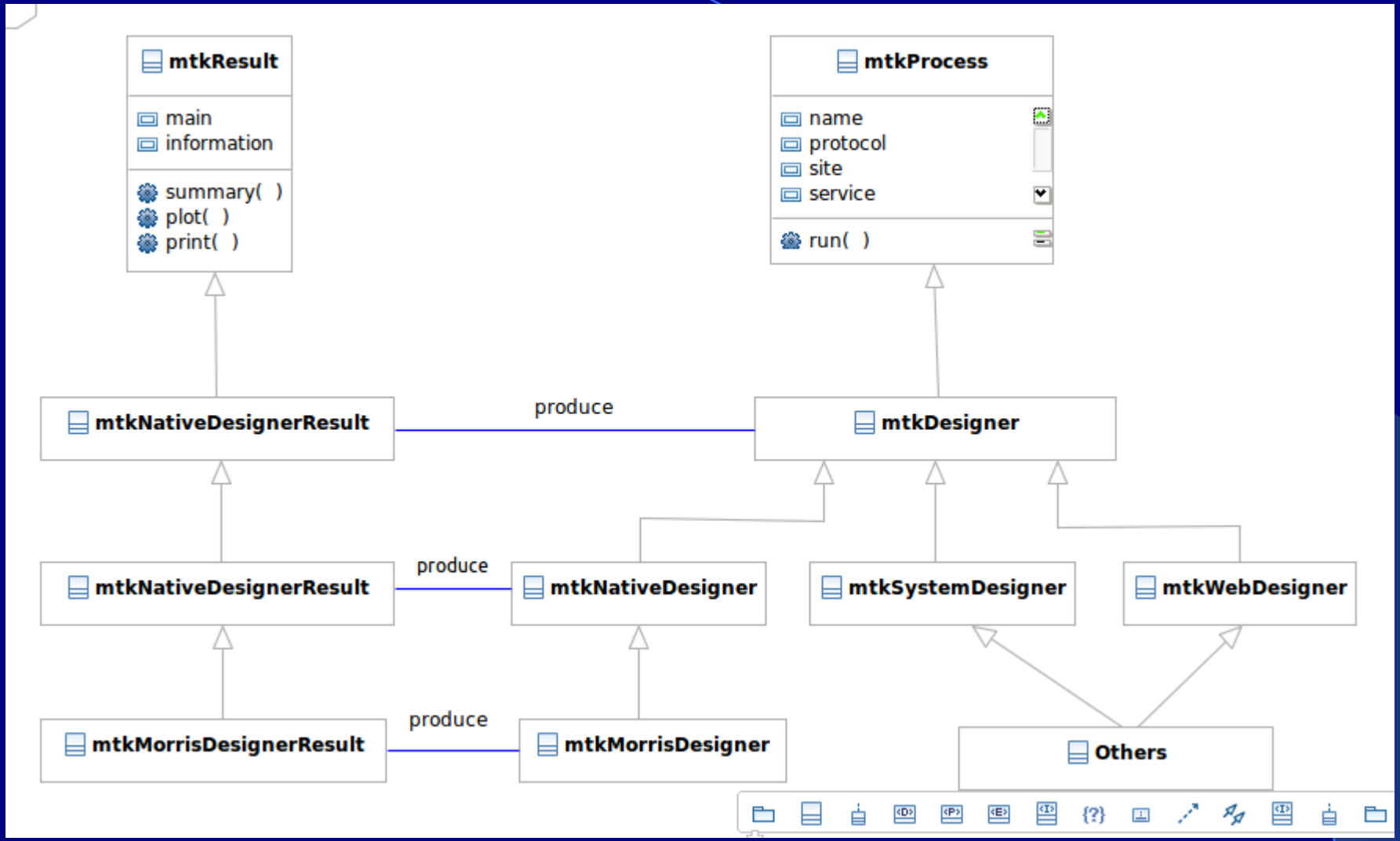
Conception (1)



Conception (2)



Conception (3)



Fonctionnalités

- Gestion flexible 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 logique de conception dans mtk

- **Décomposition en taches et recomposition**

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

- **Formalisation des tâches, de leurs gestionnaires, et des résultats produits :**
  - *design ----------> mtkDesigner ----------> mtkDesignerResult*
  - *evaluate ---------> mtkEvaluator ----------> mtkEvaluatorResult*
  - *analyze ----------> mtkAnalyser ----------> mtkAnalyserResult*

- **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*

Exemples

- **Modèle natif implémenté dans le package « mtk ».**
- **Modèle programmé comme fonction R indépendante**
- **Modèle dans une librairie ou programmés par des tiers**
- **Pilotage par des fichiers XML**
- ***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)
```

Pilotage à partir d'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="Sobol14.R",
  authors="H. Monod,INRA-MIA",
  name="Sobol14", main="Sobol14",
  plot="pl.sobol",  summary="s.sobol")
```

```
sampleur ← mtkSobol14Designer(
   listParameters = list(N=200,shrink=0.8)
   )
```