

# RDOS

## Random Discrete Object Suite

Julien David et Yann Ponty et ... vous?

ANR Magnum

13 Juin 2013

Logiciels de génération aléatoire développés dans MAGNUM/ALEA :

Logiciel	Auteur
Regal	Julien David
PSI2	Ana Busic
GenRGenS	Yann Ponty
Seed	Cyril Nicaud
Boltzmann Oracle	Carinne Pivoteau
Permutations	Dominique Rossin
GenWalks	Yann Ponty

Génération aléatoire utile dans de nombreux contextes :

- Théorie des langages (Automates...)
- Bioinformatique (ARN...)
- Visualisation (dessin de graphes/arbres)
- Génie logiciel (meta-modèles)
- Test de programme
- Programmation (XML)
- Structures de données
- ...

**Mais** implémentations peu utilisées post-publication ...

... au profit de méthodes *ad hoc*, souvent **coûteuses**, parfois même **fausses** !

Logiciels de génération aléatoire développés dans MAGNUM/ALEA :

Logiciel	Auteur
Regal	Julien David
PSI2	Ana Busic
GenRGenS	Yann Ponty
Seed	Cyril Nicaud
Boltzmann Oracle	Carinne Pivoteau
Permutations	Dominique Rossin
GenWalks	Yann Ponty

Génération aléatoire utile dans de nombreux contextes :

- Théorie des langages (Automates...)
- Bioinformatique (ARN...)
- Visualisation (dessin de graphes/arbres)
- Génie logiciel (meta-modèles)
- Test de programme
- Programmation (XML)
- Structures de données
- ...

**Mais** implémentations peu utilisées post-publication ...

... au profit de méthodes *ad hoc*, souvent **coûteuses**, parfois même **fausses** !

Logiciels de génération aléatoire développés dans MAGNUM/ALEA :

Logiciel	Auteur
Regal	Julien David
PSI2	Ana Busic
GenRGenS	Yann Ponty
Seed	Cyril Nicaud
Boltzmann Oracle	Carinne Pivoteau
Permutations	Dominique Rossin
GenWalks	Yann Ponty

Génération aléatoire utile dans de nombreux contextes :

- Théorie des langages (Automates...)
- Bioinformatique (ARN...)
- Visualisation (dessin de graphes/arbres)
- Génie logiciel (meta-modèles)
- Test de programme
- Programmation (XML)
- Structures de données
- ...

**Mais** implémentations peu utilisées post-publication ...

... au profit de méthodes *ad hoc*, souvent **coûteuses**, parfois même **fausses** !

## Obstacles côté utilisateurs . . .

- Choc culturel (Empirisme vs Mathématiques, "Bonus à la naïveté" . . .)
- Technologies/platformes diverses (Installation ?)
- Identifier un générateur adéquat : Problème de vocabulaire

## . . . et côté développeurs

- Prototype  $\neq$  logiciel
  - Diffusion logicielle (Site web, documentation)
  - Licence logicielle (Politique locale . . .)
  - Développement pérenne (URL stable, disponibilité langages/librairies)
- Entrées/sorties (Objet combinatoire : Quezako ???)
- Effort parfois insuffisant d'évangélisation

⇒ Seul un utilisateur (informaticien) persuadé de trouver son bonheur chez nous fera l'effort (compiler, modifier le code, nous contacter . . .) !

## Obstacles côté utilisateurs . . .

- Choc culturel (Empirisme vs Mathématiques, "Bonus à la naïveté" . . .)
- Technologies/platformes diverses (Installation ?)
- Identifier un générateur adéquat : Problème de vocabulaire

## . . . et côté développeurs

- Prototype  $\neq$  logiciel
  - Diffusion logicielle (Site web, documentation)
  - Licence logicielle (Politique locale ...)
  - Développement pérenne (URL stable, disponibilité langages/librairies)
- Entrées/sorties (Objet combinatoire : Quezako ???)
- Effort parfois insuffisant d'évangélisation

⇒ Seul un utilisateur (informaticien) persuadé de trouver son bonheur chez nous fera l'effort (compiler, modifier le code, nous contacter . . .) !

## Obstacles côté utilisateurs . . .

- Choc culturel (Empirisme vs Mathématiques, "Bonus à la naïveté" . . .)
- Technologies/platformes diverses (Installation ?)
- Identifier un générateur adéquat : Problème de vocabulaire

## . . . et côté développeurs

- Prototype  $\neq$  logiciel
  - Diffusion logicielle (Site web, documentation)
  - Licence logicielle (Politique locale ...)
  - Développement pérenne (URL stable, disponibilité langages/librairies)
- Entrées/sorties (Objet combinatoire : Quezako ???)
- Effort parfois insuffisant d'évangélisation

⇒ Seul un utilisateur (informaticien) persuadé de trouver son bonheur chez nous fera l'effort (compiler, modifier le code, nous contacter . . .) !

# Pourquoi pas SAGE ?



**SAGE** est un environnement logiciel mathématique généraliste, qui pourrait à terme jouer un **rôle de vitrine** pour nos méthodes.

**Mais** plusieurs obstacles rendent son adoption difficile actuellement :

- Effort supplémentaire conséquent pour intégrer nos générateurs
  - Implémentation en Python ou interfaçage C un peu complexe
  - Logique orientée objet déroutante pour certains
  - Quel(s) langage(s) de description pour nos objets ?
- Encore peu utilisé par notre (sous-)communauté (Maple)
- Problème de reconnaissance des contributions individuelles

⇒ Mise en oeuvre complexe dans la durée d'un projet ANR.

# Pourquoi pas SAGE ?



**SAGE** est un environnement logiciel mathématique généraliste, qui pourrait à terme jouer un **rôle de vitrine** pour nos méthodes.

**Mais** plusieurs obstacles rendent son adoption difficile actuellement :

- Effort supplémentaire conséquent pour intégrer nos générateurs
  - Implémentation en Python ou interfaçage C un peu complexe
  - Logique orientée objet déroutante pour certains
  - Quel(s) **langage(s) de description** pour nos objets ?
- Encore peu utilisé par notre (sous-)communauté (Maple)
- Problème de reconnaissance des contributions individuelles

⇒ Mise en oeuvre complexe dans la durée d'un projet ANR.

# Pourquoi pas SAGE ?



**SAGE** est un environnement logiciel mathématique généraliste, qui pourrait à terme jouer un **rôle de vitrine** pour nos méthodes.

**Mais** plusieurs obstacles rendent son adoption difficile actuellement :

- Effort supplémentaire conséquent pour intégrer nos générateurs
  - Implémentation en Python ou interfaçage C un peu complexe
  - Logique orientée objet déroutante pour certains
  - Quel(s) **langage(s) de description** pour nos objets ?
- Encore peu utilisé par notre (sous-)communauté (Maple)
- Problème de reconnaissance des contributions individuelles

⇒ Mise en oeuvre complexe dans la durée d'un projet ANR.

# Pourquoi pas SAGE ?



**SAGE** est un environnement logiciel mathématique généraliste, qui pourrait à terme jouer un **rôle de vitrine** pour nos méthodes.

**Mais** plusieurs obstacles rendent son adoption difficile actuellement :

- Effort supplémentaire conséquent pour intégrer nos générateurs
  - Implémentation en Python ou interfaçage C un peu complexe
  - Logique orientée objet déroutante pour certains
  - Quel(s) **langage(s) de description** pour nos objets ?
- Encore peu utilisé par notre (sous-)communauté (Maple)
- Problème de reconnaissance des contributions individuelles

⇒ Mise en oeuvre complexe dans la durée d'un projet ANR.

# Pourquoi pas SAGE ?



**SAGE** est un environnement logiciel mathématique généraliste, qui pourrait à terme jouer un **rôle de vitrine** pour nos méthodes.

**Mais** plusieurs obstacles rendent son adoption difficile actuellement :

- Effort supplémentaire conséquent pour intégrer nos générateurs
  - Implémentation en Python ou interfaçage C un peu complexe
  - Logique orientée objet déroutante pour certains
  - Quel(s) **langage(s) de description** pour nos objets ?
- Encore peu utilisé par notre (sous-)communauté (Maple)
- Problème de reconnaissance des contributions individuelles

⇒ Mise en oeuvre complexe dans la durée d'un projet ANR.

# Une solution pragmatique : La plateforme RDOS

## Random Discrete Object Suite

- Une plateforme regroupant différents générateurs aléatoires.
- Possibilité de **visualiser** les résultats en ligne, ou de les **télécharger**
- Accents sur l'**identification** des auteurs, et l'**éducation** des utilisateurs
- Possibilité de délocaliser les moyens de calcul

Projet moins ambitieux qu'une intégration SAGE, mais ...

## Avantages

- Simple d'utilisation : l'utilisateur n'installe rien
- Un développeur intègre facilement son logiciel au projet
- Amélioration immédiate de la visibilité des générateurs MAGNUM/ALEA
- Intégration facile d'un logiciel, exécuté *localement* si nécessaire
- Base de réflexion pour des formats communs

## Random Discrete Object Suite

- Une plateforme regroupant différents générateurs aléatoires.
- Possibilité de **visualiser** les résultats en ligne, ou de les **télécharger**
- Accents sur l'**identification** des auteurs, et l'**éducation** des utilisateurs
- Possibilité de délocaliser les moyens de calcul

Projet moins ambitieux qu'une intégration SAGE, mais ...

## Avantages

- Simple d'utilisation : l'utilisateur n'installe rien
- Un développeur intègre facilement son logiciel au projet
- Amélioration immédiate de la visibilité des générateurs MAGNUM/ALEA
- Intégration facile d'un logiciel, exécuté *localement* si nécessaire
- Base de réflexion pour des formats communs

# Structure de la base

Partie visible pour l'utilisateur

Site Internet

Base de données

- Liste des tâches à exécuter
- Configuration des générateurs aléatoires

Machine 1

- **Daemon RDOS**
- Logiciel A
- Logiciel B

Machine 2

- **Daemon RDOS**
- Logiciel C

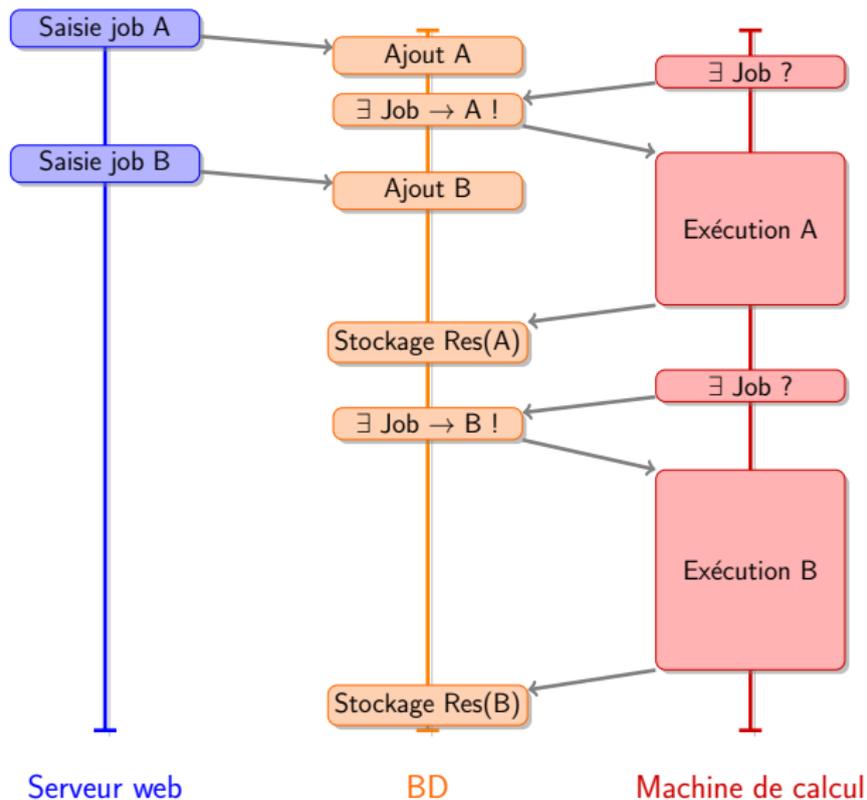
Machine 3

- **Daemon RDOS**
- Logiciel A
- Logiciel C
- Logiciel D

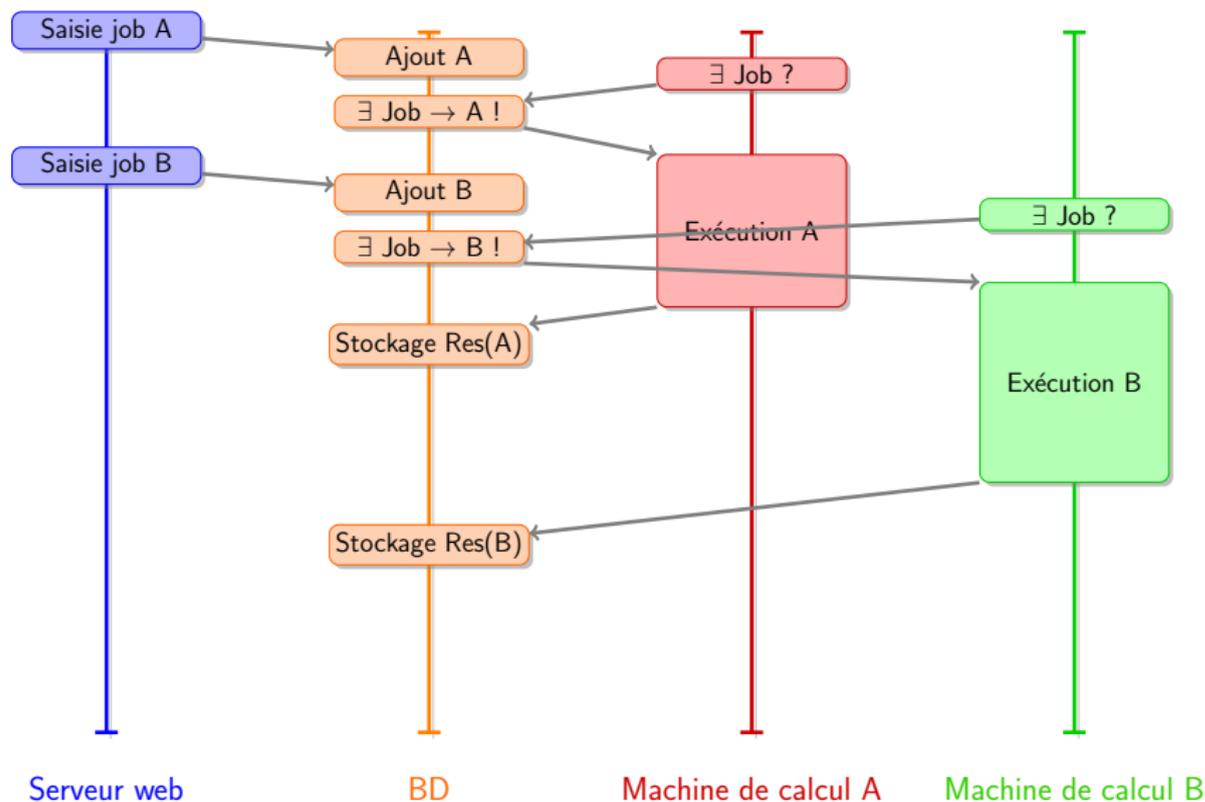
Machine 4

- **Daemon RDOS**
- Logiciel E

# Cycle de vie



# Cycle de vie



## Démonstration...

**Disclaimer** : Tout ce qui plantera aujourd'hui marchait hier (juré !).

## Prérequis

- Le programme doit être configurable sur la ligne de commande et renvoyer le résultat sur la sortie standard.

## Incorporation (par nous !)

Envoyer un mail contenant :

- Sources du programme,
- Descriptif du programme, des objets et du modèle de génération, en français et en anglais,
- Un descriptif de la ligne de commande qui permet d'appeler le programme.

## Prérequis

- Le programme doit être configurable sur la ligne de commande et renvoyer le résultat sur la sortie standard.

## Incorporation (par nous !)

Envoyer un mail contenant :

- Sources du programme,
- Descriptif du programme, des objets et du modèle de génération, en français et en anglais,
- Un descriptif de la ligne de commande qui permet d'appeler le programme.

- Intégrer plus de générateurs
- Succès *débordant*, générateurs plus coûteux et/ou basés sur des logiciels payants → Architecture distribuée
- Interconnexion autres initiatives MAGNUM/ALEA/... (ECS, BD Marne)
- Plus de visualisations

Merci à tous !  
(Merci++ à Sven et Axel)

- Intégrer plus de générateurs
- Succès *débordant*, générateurs plus coûteux et/ou basés sur des logiciels payants → Architecture distribuée
- Interconnexion autres initiatives MAGNUM/ALEA/... (ECS, BD Marne)
- Plus de visualisations

Merci à tous !  
(Merci++ à Sven et Axel)

- Intégrer plus de générateurs
- Succès *débordant*, générateurs plus coûteux et/ou basés sur des logiciels payants → Architecture distribuée
- Interconnexion autres initiatives MAGNUM/ALEA/... (ECS, BD Marne)
- Plus de visualisations

Merci à tous !  
(Merci++ à Sven et Axel)

- Intégrer plus de générateurs
- Succès *débordant*, générateurs plus coûteux et/ou basés sur des logiciels payants → Architecture distribuée
- Interconnexion autres initiatives MAGNUM/ALEA/... (ECS, BD Marne)
- Plus de visualisations

Merci à tous !  
(Merci++ à Sven et Axel)

- Intégrer plus de générateurs
- Succès *débordant*, générateurs plus coûteux et/ou basés sur des logiciels payants → Architecture distribuée
- Interconnexion autres initiatives MAGNUM/ALEA/... (ECS, BD Marne)
- Plus de visualisations

Merci à tous !  
(Merci++ à Sven et Axel)

- Intégrer plus de générateurs
- Succès *débordant*, générateurs plus coûteux et/ou basés sur des logiciels payants → Architecture distribuée
- Interconnexion autres initiatives MAGNUM/ALEA/... (ECS, BD Marne)
- Plus de visualisations

Merci à tous !  
(Merci++ à Sven et Axel)