

Yams++

SConspiracy

Construction d'application C++ en environnement contrôlé

Nicolas Philipps
nicolas.philipps@ircad.fr

IRCAD R&D
Institut de Recherche contre les Cancers de l'Appareil Digestif

European Developer Forum 5
Octobre 2009



- 1 Introduction
- 2 Génie Logiciel
- 3 SConspiracy
- 4 Mise en oeuvre
- 5 Outils externes
- 6 Conclusion

SConspiracy

- Yams++ \Rightarrow SConspiracy (racy)
- Construction d'application
 - C++
 - Multiplateforme
 - Orientée composant
 - En environnement contrôlé
- Open source, Licence BSD
 - <http://code.google.com/p/sconspiracy/>
- Compatibilité avec Yams++
- Équipe R&D IRCAD (<http://www.ircad.fr>)
- Remplace make, cmake, ...
- Python 2.6, SCons ≥ 1.2

Environnement à l'IRCAD

- Équipe de ~10 personnes
- C++
- Projets complexes
 - >250 bibliothèques/composants C++
- Multiplateforme (Linux, Mac, Windows, 32/64bits)
- Bibliothèques externes (boost, wxwidgets, vtk, ...)
- Outils complémentaires
 - Documentation (Doxygen)
 - Test unitaires (CppUnit)
 - Ferme de compilation (Hudson)

Python

- Haut niveau
- Open Source
- Multiplateforme
- Syntaxe simple, intuitive
- Boite à outils complète
- Pas de compilation explicite
- Base de SCons

SCons

- Écrit en Python, Open Source
- Construction multi plateforme/environnement/langage
- Détection de changement dans les sources
- Gestion automatique de dépendances
- Construction parallèle efficace
- Fichiers de configuration lisible

Makefile

```
HelloWorld : HelloWorld.cpp  
gcc HelloWorld.cpp -o HelloWorld
```

SCons

```
Program('HelloWorld.cpp')
```

- 1 Introduction
- 2 Génie Logiciel**
 - Méthodes agiles
- 3 SConspiracy
- 4 Mise en oeuvre
- 5 Outils externes
- 6 Conclusion

Extreme Programming

- Méthode agile
- Équipes réduites
- Principes simples poussés à l'extrême :
 - revue de code permanente
 - Cycles de développement courts
 - Intégration continue
 - Refactoring permanent
 - Tests systématiques

Extreme Programming

- Cycles de développement rapides
- Intégration continue
 - Compilations régulières par les développeurs
 - Compilations régulières par une ferme de compilation
- Tests unitaires
 - Compilations et tests par les développeurs
 - Compilations et tests automatisées (Ferme)

- 1 Introduction
- 2 Génie Logiciel
- 3 SConspiracy**
 - Buts de SConspiracy
 - Les concepts
 - Extensibilité
- 4 Mise en oeuvre
- 5 Outils externes
- 6 Conclusion

Les Buts

- Ajout/Configuration de projets souple et simple
- Respecter au maximum la philosophie DRY (Don't Repeat Yourself)
- Une seule commande pour tout construire
- Détection automatique de ce qui doit être compilé
- Obtenir un workflow de construction optimum
- Permettre facilement l'ajout ou l'évolution de fonctionnalité (plugins)
- Minimiser les modifications de configuration :
 - d'une plateforme/architecture à l'autre
 - d'un environnement de compilation à l'autre
- Évoluer dans un environnement contrôlé le plus indépendant possible du système d'exploitation

Les Buts

- Gestion uniforme et souple des bibliothèques externes
- Création de packages binaire utilisables sous forme de bibliothèques externes
- Gestion de multiples projets indépendant ou à composants commun

Terminologie

- **Projet**
- **Base de projets**
- **Bibliothèques externes**
- **Paquets binaires**

Projet

- Répertoire Structuré
- Nom du répertoire \Rightarrow nom du projet
- Sources + ressources + configuration
- Structure du répertoire
 - prj/bin/build.options
 - prj/include
 - prj/src
 - prj/rc
- build.options
 - code python
 - variable nécessaire : TYPE (shared, static, exec, bundle, ...)
 - variables optionnelles : VERSION, LIB, USE, ...

Exemple de build.options

Exemple de fichier build.options :

```
build.options
```

```
TYPE = 'shared'
```

```
VERSION = '0-4'
```

```
LIB = ['core', 'utils']
```

```
USE = ['boost', 'vtk']
```

```
[...]
```

Base de projets

- Un ou plusieurs répertoires contenant des projets
- Arborescence non structurée
- Noms de projets uniques, composé par $[a-zA-Z0-9]^+$
- Un projet ne peut pas se trouver dans le répertoire d'un autre projet

Bibliothèques externes

- Bibliothèques utilisée dans un projet (boost, wx, ...)
- Non compilée par SConspiracy
- Provenance
 - Installé sur le système
 - Compilée par l'utilisateur de SConspiracy

Paquet binaire

- Structure proche de celle d'un projet
- Plateforme/Architecture/Mode spécifique
- Module python
- Répertoire structuré
 - Description de la bibliothèque
 - Binaires + en-tête
- Peu décrire une bibliothèque du système
- Bibliothèques externes ou projets précompilés

Composants de SConspiracy

- racy
 - renv
 - rlibext
 - rplugins
 - rproject
 - rtools
 - plugins

Étapes de construction

- Création d'un environnement de compilation
- Recherche des Bibliothèques Externes disponibles
- Recherche des projets disponibles
 - Lecture des build.options
 - Création d'un project SConspiracy
- Contrôle de la satisfaction des dépendances
- Génération des cibles
- Gestion des dépendances entre les cibles
- Passage des cibles et options à SCons

Plugins

- Complète ou remplace une étape de construction
- Pas de configuration spécifique
- Fonctionnement indépendant
- Module python
- Exemple de plugins
 - Doxygen
 - CppUnit
 - Deps
 - Gcc4.4
 - FailAlert

Plugins Gcc4.4

```
# -*- coding: UTF8 -*-
```

```
import os
import racy
```

```
KEYWORD = 'USE_GCC44'
```

```
class Plugin(racy.rplugins.Plugin):
    name = 'gcc4.4'
```

```
    options          = { KEYWORD : 'no', }
    allowed_values   = { KEYWORD : ['no', 'yes'] , }
```

```
    cmdline_opts    = [ KEYWORD ]
    descriptions_opts = { KEYWORD : 'enable/disable use of gcc 4.4' , }
```

```
def has_env_addon(self, env):
    return racy.renv.options.get_option( KEYWORD ) == 'yes'
```

```
def get_env_addon(self, env):
    env.PrependENVPath('PATH', os.path.join(__path__[0], 'gcc4.4', 'bin'))
    return []
```


Exemple 1

Un exemple simple

SConstruct

```
Program('program.cpp')
```

build.options

```
TYPE = 'exec'
```

Structure

```
.../ex1/program.cpp  
.../ex1/SConstruct
```

Structure

```
.../ex1/bin/build.options  
.../ex1/src/program.cpp
```

Exemple 2

Un autre exemple simple

SConstruct

```
Program('ex2', Glob('*.*pp'))
```

build.options

```
TYPE = 'exec'
```

Structure

```
ex2/build.options  
ex2/SConspirator.hpp  
ex2/program.cpp  
ex2/SConspirator.cpp
```

Structure

```
ex2/bin/build.options  
ex2/include/SConspirator.hpp  
ex2/src/program.cpp  
ex2/src/SConspirator.cpp
```

Exemple 3 - SCons

Une application avec une bibliothèque

SConstruct

```
sconspirator = SConscript('sconspirator/SConscript')
Program('ex3', Glob('*.*.cpp'),
        CPPPATH=['sconspirator/'],
        LIBPATH=['sconspirator/'],
        LIBS =['sconspirator'],
)
```

SConscript

```
lib = SharedLibrary('sconspirator', Glob('*.*.cpp'))
Return ('lib')
```

Exemple 3 - SCons

Structure

ex3/program.cpp

ex3/sconspirator

ex3/sconspirator/SConscript

ex3/sconspirator/SConspirator.cpp

ex3/sconspirator/SConspirator.hpp

ex3/SConstruct

Exemple 3 - Racy

build.options - 'ex3'

```
TYPE = 'exec'
```

```
LIB = ['sconspirator_0.1']
```

build.options - 'sconspirator'

```
TYPE = 'shared'
```

```
VERSION = '0.1'
```

build.options - 'ex3'

```
ex3/bin/build.options
```

```
ex3/src/program.cpp
```

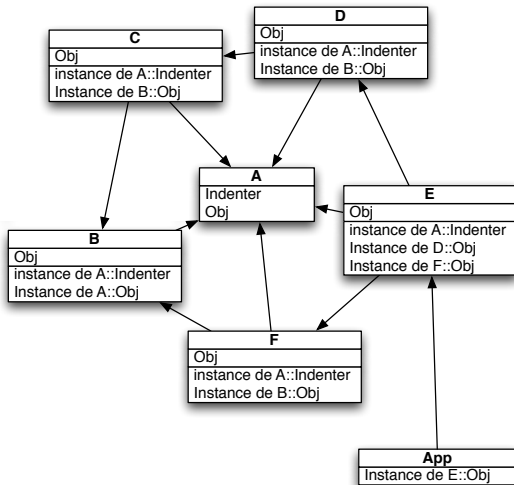
build.options - 'sconspirator'

```
sconspirator/bin/build.options
```

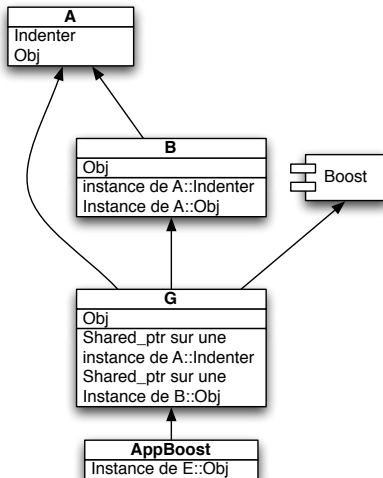
```
sconspirator/include/SConspirator.hpp
```

```
sconspirator/src/SConspirator.cpp
```

Application + libs



Application + libs + ext



- 1 Introduction
- 2 Génie Logiciel
- 3 SConspiracy
- 4 Mise en oeuvre
- 5 Outils externes**
 - CppUnit
 - Doxygen
 - Hudson
- 6 Conclusion

CppUnit

- Framework de test unitaire C++
- Open source
- avec SConspiracy
 - Plugin
 - Compilation des tests
 - bibliothèque dynamique
 - executable
 - executable + sortie XML (Hudson)
 - execution des tests

Doxygen

- Documentation de code source (C, C++, Python, Java, ...)
- Open source
- avec SConspiracy
 - Plugin
 - Génération de documentation pour un projet + ses dépendances

Hudson

- Intégration continue
- Open source
- Multiplateforme (java)
- avec SConspiracy
 - Compilation régulière, déclenchée par 'commit'
 - Execution de tests unitaires (SConspiracy/CppUnit) systématiques
 - Génération de Doxygen (SConspiracy/Doxygen) pour chaque projet

The screenshot shows the Hudson web interface in a Microsoft Internet Explorer browser window. The address bar displays 'http://localhost:8080/hudson/'. The page title is 'Hudson' and the URL is 'http://localhost:8080/hudson/'.

On the left side, there are links for 'New Job...', 'Configure...', and 'Reload Config.'. Below these are sections for 'Build Queue' and 'Build Executor Status'. The 'Build Queue' section shows a list of jobs with their status (e.g., 'Building', 'Building', 'Building', 'Building', 'Building', 'Idle').

On the right side, there is a table of build jobs with columns for 'Job', 'Last Success', 'Last Failure', and 'Last Duration'. The table lists several jobs, including 'CommonAnnotations', 'hsh', 'dot-assert', 'E', 'f. (osash)', 'sleash', 'hudson', 'lstack-common', 'JMS', 'java-net-api-community', 'java-net-api-community-updater', and 'java-net-api-processor'.

Job	Last Success	Last Failure	Last Duration
CommonAnnotations	4 days (#18)	9 months (#2)	39 seconds
hsh	6 months (#11)	10 months (#2)	59 seconds
dot-assert	6 months (#8)	N/A	1 minute
E	28 days (#50)	1 month (#507)	7 minutes
f. (osash)	6 days (#33)	13 days (#2)	5 minutes
sleash	4 hours (#10)	1 day (#20)	1 hour
hudson	4 minutes (#20)	N/A	1 minute
lstack-common	12 days (#15)	16 days (#5)	14 seconds
JMS	3 days (#33)	9 hours (#2)	1 minute
java-net-api-community	4 minutes (#1616)	10 hours (#1613)	1 minute
java-net-api-community-updater	18 hours (#36)	N/A	0 seconds

- Outil souple
- Utilisation simple
- multiplateforme
- Intégrable
- Extensible

Stages 2009-2010

<http://www.ircad.fr/applied/stages>

Sujets

- Maquette à base de technologies VoIP, VideoOnLan, visioconférence IP
- Intégration de fonctions de fusion d'images dans des logiciels médicaux
- Évaluation des frameworks C++ : Mozilla XPFE/XPCOM, Qt et FW4SPL
- Évolution du stockage d'images médicales
- Évaluation des solutions de virtualisation du marché
- Rendu Volumique : manipulation des fonctions de transfert couleur

Merci de votre attention !

IRCAD - Open Source projects

- FW4SPL : <http://code.google.com/p/fw4spl>
- SConspiracy : <http://code.google.com/p/sconspiracy>

IRCAD Freeware using SConspiracy

VR-Render 0.8 : <http://www.ircad.fr/softwares/vr-render>

IRCAD - Stages 2009-2010

<http://www.ircad.fr/applied/stages>