

 	KRIS	
	Software Quality Control & Dead Code Cleaner	
	System Requirements: Langage : C ANSI - DB : MySQL, DB2, Oracle - Serveur : Apache	
	Développé par:	Asetechs www.asetechs.com
Licence commerciale	Personne à contacter	jeanpierre.latour@smals.be

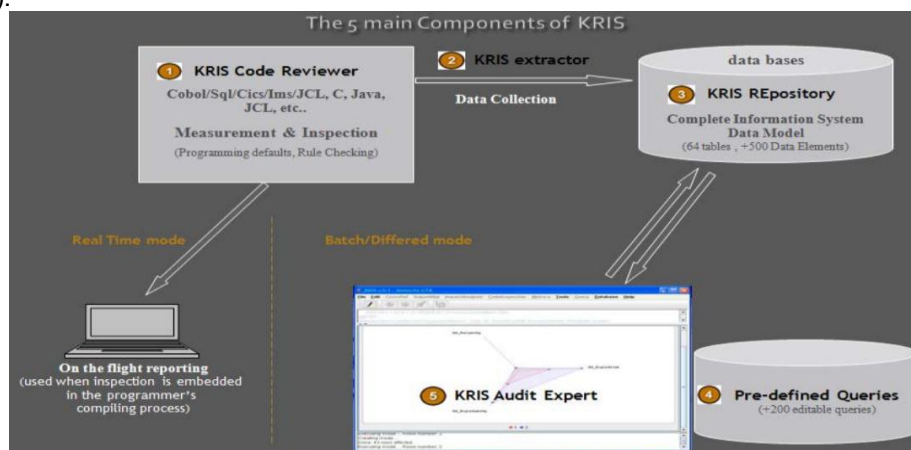
Fonctions

KRIS est une solution de software quality control. Ses trois fonctionnalités principales sont :

- le **Code Reviewer** pour l'inspection du code (vérification de l'application des règles de coding) et la mesure de la qualité résultante selon de nombreux indicateurs ;
- le **Code Normalizer** qui permet d'appliquer des transformations et des restructurations sur le code ;
- le **Dead Code Cleaner** spécialisé dans la détection du code mort et son élimination.

Ces trois modules sont évidemment complétés par un **repository** et une interface d'exploitation du contenu de celui-ci (le **KRIS Audit Expert**). Le reporting est disponible en real-time (au moment de la compilation) ou en batch.

Kris supporte les langages Cobol, C, Java, CICS, IMS, JCL(IBM), MFS, BMS, PROC, PARM, SQL (DB2, Oracle) Natural, Adabas, PL1 et Asetechs a développé des parsers pour de nombreux dialectes (XML, HTML, REXX, Cshell, ...).



A titre d'exemples, voici quelques mesures prises par le code reviewer : le nombre de lignes de code, de commentaires, d'interlignes, d'instructions, d'instructions mortes, d'opérateurs, d'opérandes, la complexité cyclomatique (McCabe), le nombre de violations des règles de codage définies, ... KRIS supporte l'analyse statique mais pas dynamique. Au total 46 métriques sont supportées.

Sur base de ces mesures, le logiciel calcule un ensemble d'indicateurs tels que la qualité, la maintenabilité, la fiabilité, la portabilité, le ratio de code mort, ... Il sera aussi utile pour évaluer les impacts des défauts structurels et des dépendances inter-composants : l'outil permet de détecter toutes les chaînes d'inputs et d'outputs à partir d'un module de programmation donné.

Les deux premiers modules sont disponibles pour les langages Cobol et Java, le dernier uniquement pour les environnements Cobol/CICS/IMS/SQL. D'autres langages sont éligibles : PL1, C, PL/SQL, JCL, Natural Fortran, ...

Signalons encore que l'outil permet les analyses d'impact.

Fonctions (suite)

Le **dead code analyser** a particulièrement retenu notre attention, étant donné son utilité potentielle dans le cadre d'opérations de migration de systèmes legacy. Le code mort représente en moyenne 15 % du total du code d'une application, et peut monter à plus de 30% dans les très vieux systèmes.

Le dead code analyser localise le code et les variables mortes, ainsi que les références à des copybooks qui ne sont plus utilisés. La suppression se fait par le biais de la mise en commentaires ou par effacement.

L'intérêt de l'outil peut se justifier par:

- l'impossibilité pratique de localiser et supprimer le code et les variables mortes par la voie manuelle ;
- le fait que le code mort croît tout au long de la vie d'une application ;
- le fait que l'élimination du code mort est une source récurrente de réduction des coûts de maintenance logicielle (voir à cet égard le calcul qui suit).

Selon un calcul de l'éditeur (que nous ne reproduisons pas ici par manque de place), le gain en coûts de maintenance sur 5 ans pour un patrimoine applicatif de 5 millions de lignes de code (Cobol), se chiffrerait à plus de 1,8 million d'Euros, si l'on valorise la ligne de code à +/- 0,5 euros. Ceci est une estimation très basse : en effet, le montant généralement avancé se situe plutôt autour de 3 euros sinon davantage. Il ne semble donc pas que l'économie potentielle soit surévaluée.

Conclusions / Recommandations

Dans le cadre des opérations de migration, tout particulièrement si l'on souhaite recourir à l'automatisation, il est selon nous impératif de commencer par vérifier la qualité du code et de le débarrasser de ses branches mortes. Vouloir automatiser la conversion d'un « mauvais » code ne pourra conduire qu'à un résultat médiocre, en emportant dans le futur toute la dette technique du passé. Le code qui ne satisfait pas à un seuil minimum de qualité et de propreté doit faire l'objet d'une réécriture manuelle, opération pour laquelle le nettoyage des éléments morts a aussi tout son sens. En effet, la qualité des documentations étant habituellement ce qu'elle est, éliminer le code mort ne pourra être que profitable lorsqu'il s'agit de "plonger" dans le code existant pour en redécouvrir la finalité. On évite ainsi le temps perdu à la redécouverte de code inutile, et on élimine le parasitage sur la compréhension du code utile par le code inutile.

Il faut voir aussi que la facturation pour une migration automatisée est généralement basée sur le nombre de lignes de code, avec un taux unitaire déterminé dans une phase de pré étude ... qui a notamment pour but de déterminer la qualité et la propreté du code à migrer. L'outil sera aussi utile pour déterminer le nombre d'instructions « exotiques » qui demanderont un effort de migration spécifique.

Tests et Résultats

Nous n'avons pas à ce jour mené d'expérimentations sur le logiciel, mais le considérons avec intérêt dans le cadre de plusieurs opérations de migration à prévoir au futur.

Il nous paraît aussi bon d'insister sur le fait, qu'alors que la mise en œuvre d'outils de contrôle de la qualité du code sur Java tend à prendre de l'ampleur, ils sont souvent trop peu répandus dans les environnements Cobol.

Budget

Le dénombrement du code mort est gratuit (via une version téléchargeable).

L'utilisation de KRIS pour l'analyse détaillée et la correction automatisée est au volume (prix en février 2012):

- De 200.000 à 1 million de lignes utiles: 2 centimes d'Euro par ligne
- De 1 million à 2 millions de lignes utiles: 1,75 centimes d'Euro par ligne
- De 2 millions à 5 millions de lignes utiles: 1,50 centimes d'Euro par ligne
- Au delà de 5 millions de lignes traitées: 1 centime d'Euro par ligne
- Petit patrimoine de 500.000 lignes: 10 000 EUR

Le produit est distribué et supporté par la firme Telebig, spécialisée dans les opérations de migration mainframe. Le marché primaire concerne les pays francophones.