

# Interface utilisateur d'OASIS4

Jean-Marie Epitalon  
Novembre 2010

Rapport technique CERFACS  
TR-CMGC-09-101

# **Première partie : développement d'un prototype**

## **Introduction**

Un prototype réalisé en Tcl/Tk sur le modèle de l'interface utilisateur du coupleur PALM.

12 semaines de travail réparties comme suit :

- Phase 1 : Analyse du problème et présentation de solutions fonctionnelles : 3 semaines
- Phase 2 : Proposition d'implémentation technique de la solution : 3 semaines
- Phase 3 : Réalisation d'une version beta de l'outil : 6 semaines

ce qui correspond point par point aux annexes du contrat entre le CERFACS et J.M. EPITALON. Le présent document est la conclusion de la troisième partie du travail : le développement d'un prototype d'un logiciel. Il sert aussi de rapport final pour l'ensemble du travail.

Les deux premières parties du travail ont été conclues par l'écriture respectivement des deux documents suivants :

- « Description fonctionnelle de l'interface utilisateur d'OASIS 4 ».
- « Rapport technique d'implémentation de l'interface utilisateur d'OASIS 4 »

Pendant la première partie, on a aussi étudié les fonctionnalités d'un autre outil destiné aux utilisateurs d'OASIS4 : un outil permettant de générer automatiquement les fichiers de définition des composants du couplage. Cette étude a résulté dans l'écriture du document intitulé « Générateur automatique de descriptions XML d'application et de composants (GADAC) ».

## **Compte rendu du développement du prototype**

### **1.A) Résultats obtenus**

#### **a) Résultat général**

Le résultat du développement est un logiciel prototype conforme aux spécifications élaborées dans la phase 1.

Il est fortement inspiré de l'interface de PrePALM. Il permet à l'utilisateur d'OASIS de définir et de paramétrer un graphe de couplage avec de simples clics de souris et quelques saisies au clavier.

#### **b) Fonctionnalités**

Les fonctions principales sont présentes dans le prototype : avec ce logiciel, on peut presque déjà générer entièrement le SCC et les SMIOC d'un couplage.

Il manque cependant quelques fonctions indispensables :

- sauvegarde du schéma de couplage pour reprise ultérieure de sa conception
- saisie des paramètres généraux du couplage
- saisie des paramètres généraux d'une application
- la mise à jour d'un composant sur le graphe de couplage

Bien entendu, quelques détails de l'interface sont à revoir ou à compléter car ce logiciel n'est qu'un prototype. Par exemple, la vérification de cohérence des données saisies n'est pas faite entièrement.

Le graphisme et la convivialité de l'interface ont été peaufinés par plusieurs étapes d'évaluation : l'aspect graphique général semble satisfaisant.

## **1.B) Technique utilisée : le langage Tcl/Tk**

Le logiciel est entièrement basé sur le langage Tcl/Tk (Tool Command Language/ Tool-kit). Il est écrit dans ce langage et utilise des bibliothèques open-source, écrites en Tcl ou en langage C, mais adaptées au langage Tcl.

Le langage Tcl, adjoint de son tool-kit Tk, est un langage de script, très souple, adapté principalement à deux types de logiciel :

- traitement de données textuelles (Tcl)
- interface utilisateur graphique (Tk)

Pour l'accès aux fichiers XML, le logiciel utilise des modules externes open-source : libxml2, TclXML et TclDOM. Ces modules facilitent beaucoup la lecture et l'écriture de fichiers XML depuis un script Tcl.

Le langage Tcl/Tk est né dans le monde Unix. Tout ordinateur, personnel ou station de travail, contient un environnement Tcl/Tk immédiatement utilisable. Par contre, et ceci peut être vu comme un défaut, il est possible d'utiliser Tcl/Tk sur un poste équipé de Microsoft Windows mais cela demande l'installation préalable d'un logiciel gratuit.

En conclusion, ce langage est un bon choix. On peut quand-même souligner que d'autres options étaient envisageables :

- le couple Python/Tk
- le langage Java et une interface de type Web

Rappelons que le choix du langage Tcl/Tk vient du logiciel PrePALM.

## **1.C) Difficultés rencontrées**

Le développement du prototype a été laborieux pour les raisons suivantes :

1. On est parti du logiciel PrePALM qui a des fonctions similaires au prototype voulu mais a aussi beaucoup de fonctions supplémentaires ou légèrement différentes. Il a fallu analyser le code source de PrePALM pour en extraire les fonctions intéressantes. Du fait de la longueur et de la complexité du code source, cette analyse ne s'est pas faite sans difficulté.
2. On n'était pas du tout familier avec le langage Tcl/Tk. Il a fallu en apprendre toutes les détails et toutes les finesses. On l'a fait « sur le tas ».
3. On a perdu du temps en cherchant à programmer des accès aux fichiers NetCDF directement depuis le langage Tcl. Cette solution, quoique faisable, n'a pas été retenue car elle introduisait une trop grande complexité à l'installation des modules externes nécessaires. La solution retenue se base sur le petit programme utilitaire *ncdump* fourni avec la bibliothèque NetCDF, que tout utilisateur d'OASIS à priori a déjà en sa possession.

## **1.D) Durée des développements**

A cause des difficultés rencontrées, la durée des développements a atteint exactement six semaines. Mais le prototype n'est pas totalement opérationnel.

## **1.E) Conclusion**

La durée prévue pour le développement du prototype a été légèrement sous-estimée. Le résultat est cependant globalement satisfaisant.

On peut se poser la question suivante: les fonctions sont héritées de PrePALM; sont elles vraiment adaptées aux utilisateurs d'OASIS4 ? Pour y répondre, on attendra un retour des premiers utilisateurs.

Tcl/tk est très puissant : on peut imaginer une interface encore plus conviviale moyennant des

développements supplémentaires.

## **Travail restant à faire**

Voici la liste des fonctionnalités à ajouter avec en regard, une estimation du temps de travail de développement nécessaire.

### **1.F) Sauvegarde d'un graphe de couplage**

Ca consiste à sauver dans un fichier les données du graphe de couplage afin de pouvoir retravailler ce graphe à un moment ultérieur.

- sauvegarde des données graphiques dans un fichier XML : 1 jour
- rechargement en mémoire et sur le graphe des composants et de leur connexions à partir de ce fichier, du fichier SCC et des fichiers SMIOC : 4 jours

### **1.G) Mise à jour d'un composant**

Ca consiste à recharger en mémoire et sur le graphe de couplage un composant qui peut avoir été modifié entre temps.

- > mise à jour du composant sur le graphe avec ses nouveaux points de couplage
- estimation : 2 jours

### **1.H) Vérifications de cohérence**

- composants optionnels : avertissement si suppression de composants non optionnels : 2 heures
- changement de grille --> vérification d'un remailage sur une connexion : 1 jour
- vérification de la période des échanges (plus petit commun multiple) sur une connexion : 4 heures

### **1.I) Saisie de paramètres**

- saisie de paramètres généraux (ex : date départ simulation, nom de la machine exécutante, etc.) : 1 jour
- saisie de paramètres des applications : 4 heures
- changement de couleur des composants et fichiers : 1 heure
- suffixe pour les fichiers en Input : pouvoir saisir le nom du fichier

### **1.J) Installation**

- procédure d'installation facile : un script shell qui
  - installe les modules Tcl utiles (TclXML, TclDOM)
  - copie les autres fichiers dans un répertoire local
- estimation : 3 jours

### **1.K) Aide**

- aide contextuelle par clic droit sur les champs de saisie : 1 jour

- affichage d'une légende pour les pictogrammes de transformation : 2 heures

### **1.L) Suggestions**

- afficher une info-bulle sur les pictogrammes des transformations : 3 heures
- afficher l'axe des temps des fichiers input : 4 heures
- possibilité de déplacer les fils de connexion sur le graphe : 3 jours
- une option permettant de cacher les pictogrammes des transformations

### **1.M) Estimation de la durée totale de développement**

~ 18 jours

## **Projets pour la suite**

Il s'agit de faire le travail évalué au paragraphe précédent afin de rendre le logiciel prototype opérationnel et agréable d'utilisation.

D'autre part, le développement d'un autre logiciel d'interface est envisagé :

un logiciel permettant la génération des fichiers XML de définition des composants du couplage : PMIOD et AD. Ce logiciel offrirait une interface graphique posant des questions à l'utilisateur.

Ceci veut dire qu'on a abandonné l'idée de développer un logiciel générant automatiquement ces fichiers à partir des sources Fortran des composants.

## **Deuxième partie : phase terminale du développement**

### **Introduction**

La phase terminale de développement en question a compris trois volets :

- La première phase de développement a porté sur la création d'un prototype de logiciel d'interface graphique pour utilisateur d'OASIS 4. Ce logiciel, écrit en Tcl/Tk, permet la génération automatique des fichiers de configuration d'un couplage à partir de saisies au clavier et à la souris. Ce prototype nécessitait quelques améliorations.
- Le logiciel étant composé de plusieurs unités logicielles, son installation sur un ordinateur était assez complexe. Il s'agissait de rendre cette installation plus facile en l'automatisant.
- Enfin, il s'agissait de développer un autre outil logiciel pouvant servir à un utilisateur d'OASIS 4 dans sa phase amont de spécification. Ce deuxième logiciel devait servir à générer les fichiers de description d'une application (de modélisation du climat) et de ses composants.

Ce document a pour but de décrire chacune de ces trois phases de développement, leur résultat et les techniques utilisées. Il termine par les fonctions qui seraient souhaitables d'ajouter à ces deux outils logiciels.

### **1) Interface de configuration de couplage**

Il était proposé d'améliorer le prototype en ajoutant les fonctions suivantes :

- recharge en mémoire et sur le graphe de couplage d'un composant qui peut avoir été modifié entre temps
- reprise d'une configuration de couplage préexistante pour en faire une nouvelle version
- vérifications de cohérence des choix de l'utilisateur (composants optionnels, remaillage obligatoire en cas de grilles source et cible différentes, période des échanges, etc.)
- saisie des paramètres généraux du couplage (dates de simulation, plateforme de calcul, etc.)
- aide contextuelle par clic droit sur les champs de saisie

#### ***1.N) Recharge d'un composant qui peut avoir été modifié entre temps***

L'interface permet maintenant à un utilisateur de changer les fichiers de description d'application et de composants en même temps qu'il construit son graphe de couplage.

L'utilisateur peut mettre à jour un fichier de description puis demander à son logiciel de configuration de couplage de prendre en compte la nouvelle version.

#### ***1.O) Reprise d'une configuration de couplage préexistante***

Le logiciel permet maintenant de relire un fichier de configuration de couplage qu'il a lui-même généré à une date précédente, à condition d'avoir aussi accès aux fichiers de description correspondants aux applications du couplage. Un utilisateur peut donc générer facilement plusieurs versions successives d'un couplage.

#### ***1.P) Vérifications de cohérence des choix de l'utilisateur***

Cette fonction a été améliorée : on a en particulier fait attention à ne pas laisser l'utilisateur oublier

de saisir des données indispensables. Plutôt que d'empêcher l'utilisateur de saisir des données incohérentes, ce qui souvent indispose l'utilisateur, on a pris soin d'initialiser les champs de saisie avec des valeurs standard et cohérentes entre elles.

Il manque à cette fonction encore quelques points :

- vérifier que les champs de couplage indiqués comme « required » soient bien connectés à une source de donnée
- autoriser l'utilisateur à enlever du graphe de couplage des composants d'application optionnels
- lors de l'établissement d'une connexion entre deux champs de couplage, initialiser la période des échanges à une valeur rationnelle : la plus grande des deux périodes minimales.

### **1.Q) Saisie des paramètres généraux du couplage**

Cette fonction était indispensable pour générer une configuration de couplage complète. Elle a été ajoutée.

### **1.R) Aide contextuelle**

On a ajouté une fonction d'aide contextuelle sous la forme de bulles d'aide qui expliquent chaque champ de saisie, tout au long du parcours de l'utilisateur dans les formulaires de saisie.

## **2) Programme d'installation**

L'interface graphique de configuration de couplage est écrite en Tcl/Tk. Elle fait appel à des modules externes open-source eux-mêmes écrits en C et en Tcl/Tk, en particulier pour l'accès aux fichiers XML. L'installation de ces modules externes n'est pas simple du fait qu'ils sont écrits partiellement en C et que leur installation nécessite une phase de compilation.

Nous avons rassemblé l'ensemble des fichiers nécessaires à l'installation dans une 'archive' à laquelle nous avons ajouté un script automatique d'installation que nous avons développé. Nous avons aussi écrit une documentation qui accompagne le script d'installation.

Ce script simplifie la procédure d'installation. Il est écrit en Tcl. Il fait appel à des programmes exécutables qui sont présents sur tous les systèmes Unix/Linux. Il est donc directement utilisable sur tous ces systèmes.

Il arrive cependant que certains systèmes ne soient pas standard. Pour ce cas, nous avons écrit une procédure manuelle d'installation qui reprend et décrit une à une les étapes de l'installation automatique et les paramètres possibles.

## **3) Génération automatique des descriptions d'application**

La configuration d'un couplage avec OASIS 4 fait intervenir plusieurs fichiers de configuration au format XML :

- en premier lieu, les modèles et leurs entrées/sorties doivent être décrits : c'est ce qu'on appelle les fichiers de description d'application et de composants d'application
- à partir de ces fichiers, l'utilisateur d'OASIS 4 doit créer un fichier de configuration de couplage, qui décrit les couplages entre les modèles et des paramètres relatifs au 'run' de couplage.

La deuxième étape était automatisée par l'outil logiciel créé en premier : celui qu'on nomme dans ce document « interface de configuration de couplage » et dont on parle dans le premier paragraphe.

Il s'agissait d'automatiser la première étape.

## 1.S) Fonctions

Dans cette étape, un ensemble de fichiers au format XML est automatiquement généré sur la base de données renseignées au clavier par l'utilisateur dans plusieurs formulaires présentés successivement à l'écran.

Deux types de fichiers sont générés :

4. un fichier de description d'application
5. des fichiers de description de composants d'application

Ces fichiers contiennent des données reflétant la structure des programmes des modèles : des choix fonctionnels (champs de couplage) ou structurels (multi-processing) faits par les développeurs de ces programmes.

Typiquement, la création de ces fichiers de description serait à la charge des développeurs des modèles. Cependant, grâce à cet outil et grâce à la documentation d'OASIS 4, une autre personne peut faire ce travail. Cette personne doit auparavant prendre connaissance du code source des modèles.

## 1.T) Détail des fonctions

Cet outil logiciel adresse une seule application à la fois. Il génère donc un fichier de description pour une application (AD ou Application Description) et un fichier de description pour chaque composant de l'application (PMIOD ou Potential Model Input/output description).

Il présente à l'écran une suite de formulaires à remplir :

- un formulaire général pour l'application
- un formulaire pour chaque composant de l'application
- un formulaire pour chaque champ de couplage des composants

Après la saisie d'un formulaire, les données sont vérifiées par le programme avant de passer au formulaire suivant. A tout moment, l'utilisateur peut revenir à un écran précédent pour corriger une donnée saisie.

A l'écran final, le programme affiche la liste des fichiers qui vont être générés. L'utilisateur peut alors valider ses données et lancer la génération automatique des fichiers de description.

## 1.U) Technique

Cet outil logiciel est écrit en Tcl/Tk comme le premier. Il fait appel aux mêmes unités logicielles open-source pour l'accès aux fichiers XML.

A la différence du premier outil, il ne présente pas les données graphiquement.

## 1.V) Aide de l'utilisateur

Ce programme présente une bulle d'aide contextuelle pour chaque champ à saisir par l'utilisateur.

Il initialise chaque champ avec une valeur standard raisonnable et vérifie que l'utilisateur a saisi tous les champs indispensables.



## **4) Améliorations souhaitables**

### **1.W) Outil de configuration de couplage**

Ayant déjà passé deux phases de développement, cet outil est à peu près au point. Il passera au test des utilisateurs, et ceux ci feront certainement des remarques constructives en ce qui concerne l'interface.

On a identifié déjà un point à améliorer : les vérifications de cohérence que le programme peut faire.

### **1.X) Outil de génération de descriptions**

On identifie les améliorations possibles suivantes :

#### **a) Le fichier des noms de champs standards CF**

L'outil doit lire le fichier des noms de champs standards CF : ce fichier est pour l'instant fourni avec l'outil et c'est à la charge de l'utilisateur de le mettre à jour si besoin.

Les nouvelles versions étant disponibles sur le site web du laboratoire PCMI en Californie, l'outil pourrait le télécharger automatiquement si une nouvelle version est disponible.

La difficulté est la suivante : l'URL d'accès au fichier qui est disponible sur le site web du PCMDI n'est pas constante d'une version à l'autre du fichier. En effet, l'URL est composée avec le numéro de version.

Cette difficulté mise à part, l'autre point important est que le fichier du PCMDI est au format XML et l'outil, pour l'instant, ne lit qu'un fichier texte.

#### **b) Recherche d'un nom dans la liste des champs standards CF**

Cette liste est très longue. Il est important d'aider l'utilisateur à choisir un élément de la liste. Nous avons créé une fonction permettant la recherche d'un nom par la saisie d'un mot clef, mais cette fonction est incomplète car elle s'arrête au premier no