4

Utilisation avancée

Dans ce chapitre, nous commencerons par vous présenter les structures de données manipulées par Xcos, dont la compréhension est indispensable pour une maîtrise avancée de Xcos. Après cette courte section préliminaire, nous vous exposerons dans un premier temps comment analyser finement le déroulement de la simulation pour s'assurer de la justesse de son modèle. Puis, nous verrons comment exploiter l'environnement Scilab/Xcos pour réaliser, par exemple, des analyses locales par linéarisation, caler ses modèles par optimisation paramétrique, mais aussi pour générer les codes C permettant d'utiliser ces modèles dans d'autres applications voire dans des systèmes embarqués.

4.1. Principales structures de données des schémas

Pour un usage avancé de Xcos, il est utile de connaître les principales structures de données utilisées par celui-ci pour représenter et simuler les schémas.

La structure *scs_m*

Cette structure de données contient l'ensemble de l'information associée à un schéma : les structures `sciblk` associées à chacun des blocs, la définition des liens, le contexte, etc. C'est une `tlist` Scilab de type `diagram` dont une description complète est donnée par la page d'aide de `scicos_diagram` (taper `help scicos_diagram` dans la console). Il s'agit d'une structure arborescente où chaque super bloc est lui-même représenté par une structure `scs_m` stockée dans le champ `rpar` du bloc. Le menu Affichage/Navigateur de diagrammes permet de visualiser le contenu de cette structure.

Cette structure, utilisée en interne par l'éditeur Xcos, peut être chargée dans Scilab par la fonction `importXcosDiagram`. Cette fonction prend comme argument d'entrée le nom du fichier dans lequel le schéma a été sauvegardé et génère la variable de nom `scs_m` dans l'environnement Scilab.

Cet extrait provient du livre *Scilab : De la théorie à la pratique – II. Modéliser et simuler avec Xcos* écrit par Yvon Degré et Serge Steer – © 2014 Éditions D-BookeR

La structure *%cpr*

Cette structure de données est produite par le compilateur de Xcos, elle contient toutes les informations nécessaires à la simulation, liste des fonctions de simulation, paramètres, états, tables d'ordonnancement, etc.

C'est une `tlist` Scilab de type `cpr`. Le champ *`sim`* de cette structure contient les données statiques du modèle tels que le nom des fonctions de simulation (champ *`fun`*), les paramètres réels de l'ensemble des blocs (champ *`rpar`*), etc. Le champ *`state`* contient les données du modèle qui peuvent évoluer durant la simulation.

Une description complète de ces structures de données est fournie par les pages d'aide de `scicos_cpr`, `scicos_sim` et `scicos_state`.

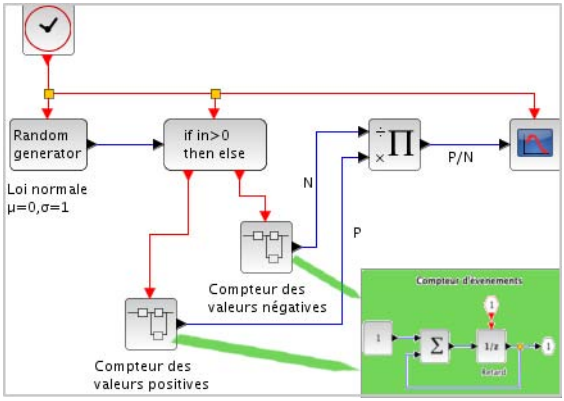
Cette structure est le plus souvent produite par le compilateur avant le lancement de la compilation et n'est pas directement accessible par l'utilisateur. Il est toutefois possible de l'obtenir à partir de la structure *`scs_m`* en utilisant l'instruction `%cpr=xcos_compile(scs_m);`.

À partir de la version Scilab 5.5.0, la fonction `xcos_blocks_info` permet de générer un tableau de chaînes de caractères qui fournit pour chaque bloc son index dans la structure compilée, son identificateur unique ainsi que les plages d'indices de ses paramètres et de ses états dans les champs correspondant de la structure *`%cpr`*. Cette table permet en particulier d'associer les numéros des blocs dans la structure compilée aux blocs présents dans le schéma et de localiser les paramètres et les états de chacun des blocs dans les champs correspondants de la structure compilée.

Exemples de structures de données *scs_m* et *%cpr*

Pour illustrer cette présentation des structures de données associées à un schéma, nous allons les examiner pour le schéma ci-dessous (scilab-xcos-discret.zcos) qui a été présenté au chapitre Prise en main.

Cet extrait provient du livre Scilab : De la théorie à la pratique – II. Modéliser et simuler avec Xcos
écrit par Yvon Degré et Serge Steer – © 2014 Éditions D-BookeR



Exemple 4.1 : *La structure de données scs_m*

```
-->importXcosDiagram("scilab-xcos-discret.zcos") ❶
        -->scs_m ❷
        scs_m  =
        wpar = [600,450,0,0,600,450]
        title = "scilabxcosdiscret"
        tol = [0.000001;0.000001;1.000D-10;100001;0;1;0]
        tf = 50000
        context = []
        void1 = []
        void2 = []
        void3 = []
        doc = list()
        1     RAND_m
        2     IFTHEL_f
        3     SPLIT_f
        4     SUPER_f
        5     CLOCK_c
        6     PRODUCT
        7     SUPER_f
        8     CSCOPE
        9     CLKSPLIT_f
        -->scs_m.objs(6) //Le bloc PRODUCT ❸
        ans  =

        GUI     : PRODUCT
        Graphics:  ❹
        …
        Model   :
        …

        -->scs_m.objs(7).model.rpar ❺
        ans  =
```

Cet extrait provient du livre Scilab : De la théorie à la pratique – II. Modéliser et simuler avec Xcos
écrit par Yvon Degré et Serge Steer – © 2014 Éditions D-BookeR

```
wpar = [600,450,0,0,600,450]
title = "Compteur_desbrvaleurs_positives"
tol = [0.0001,0.000001,1.000D-10,100001,0,0,0]
tf = 100000
context = ""
void1 = []
void2 = []
void3 = []
doc = list()
1     BIGSOM_f
2     DOLLAR_f
3     SPLIT_f
4     OUT_f
5     CLKINV_f
6     CONST_m
```

❶ Cette instruction charge le schéma qui a été sauvegardé dans le fichier `scilab-xcos-discret.zcos` et crée la variable *scs_m* dans l'environnement Scilab.

❷ Cette instruction visualise le contenu de la variable `scs_m` sous une forme abrégée.

❸ Les structures de données *sciblk* des blocs du schéma sont stockées dans la liste désignée par le champ *objs*.

❹ La visualisation des champs *graphics* et *model* du bloc PRODUCT a été supprimée.

❺ La structure de données *scs_m* d'un super bloc est contenue dans le champ *model.rpar* du bloc.

Exemple 4.2 : *La structure de données %cpr*

```
-->%cpr=xcos_compile(scs_m); ❶
-->fieldnames(%cpr)' ❷
   ans  =
        !state  sim  cor  corinv  !
-->%cpr.sim.rpar ❸
   ans  =

        0.
        1.
        1.
        1.
        1.
        0.1
        0.1
        1.
        1.
        1.
        0.
        0.9
        1.1
```

Cet extrait provient du livre Scilab : De la théorie à la pratique – II. Modéliser et simuler avec Xcos
écrit par Yvon Degré et Serge Steer – © 2014 Éditions D-BookeR

```
        50000.

-->%cpr.state.z ❹
   ans  =

        2113248.
        0.
        1.
        1.

-->T=xcos_blocks_info(%cpr);editvar("T"); ❺
```

❶ Cette instruction compile le schéma représenté par la structure de données *scs_m*.

❷ Les champs de la structure `%cpr`.

Les champ `corinv` et `cor` sont des listes Scilab qui permettent, connaissant le numéro d'un bloc dans la structure `%cpr.corinv(n)`, de retouver son instance dans la structure *scs_m* définissant le schéma et réciproquement (voir la page d'aide de scicos_cpr pour plus de détails).

❸ Le tableau contenant la concaténation des champs *rpar* de chacun des blocs.

❹ Le tableau contenant la concaténation des états discrets de chacun des blocs.

❺ Visualisation sous forme de table des données relatives aux blocs dans la structure compilée.

block num...	simulation...	uid	rpar index	ipar index	opar index	z index	oz index	x index
1	rndblk_m	3a6a413...	1:2	1:1		1:2		
2	ifthel	3a6a413...						
3	sum	3a6a413...	3:4					
4	dollar	3a6a413...				3:3		
5	cstblk4	3a6a413...	5:5					
6	evtdly4	-6b1024...	6:7					
7	product	-6b1024...		2:3				
8	sum	-6b1024...	8:9					
9	dollar	-6b1024...				4:4		
10	cstblk4	-6b1024...	10:10					
11	cscope	47e948f4...	11:14	4:18				

4.2. Contrôle et débogage des schémas

Il est souvent difficile de savoir si le schéma réalisé correspond effectivement à ce que vous souhaitez modéliser, tout particulièrement pour les systèmes discrets multi-horloges ou les systèmes hybrides. Pour être à même d'analyser le comportement d'un schéma, vous devez bien en maîtriser les aspects sémantiques détaillés au chapitre Signification des composants graphiques. Xcos fournit des outils permettant d'observer la séquence

Cet extrait provient du livre Scilab : De la théorie à la pratique - II. Modéliser et simuler avec Xcos
écrit par Yvon Degré et Serge Steer - © 2014 Éditions D-BookeR