

Présentation : 25/07/2014

*Stage facultatif sur la génération de code parallèle pour les langages synchrones
à l'IRIT avec l'équipe ACADIE*

In : Fichier Signal

```

type integer;
type boolean = enum (true, false);
type test = enum (un, deux, trois);

process P = (? integer a; boolean b; ! integer u, v, x, y, z;)(
|x := a when b
|y := x +1
|z := a +3
|u := (y $1 init 0) default 0
|v := (y $1 init 1) default 1
)end;

```

analyse syntaxique

Specification

process_list

header

process_name

signal_declarations

input_signal_list

output_signal_list;

local_signal_list

local_process_list

body

assignment_list

assigned_signal_name

signal_expression

constraint_list

constraint_kind

left_signal_name

right_signal_name

instantiation_list

instance_process_name

instance_output_signals

instance_input_expressions

type_declaration_list

procedure_declaration_list

Ter_chk_spec

Ter_no_submod

Ter_arith_to_call

Ter chk spec

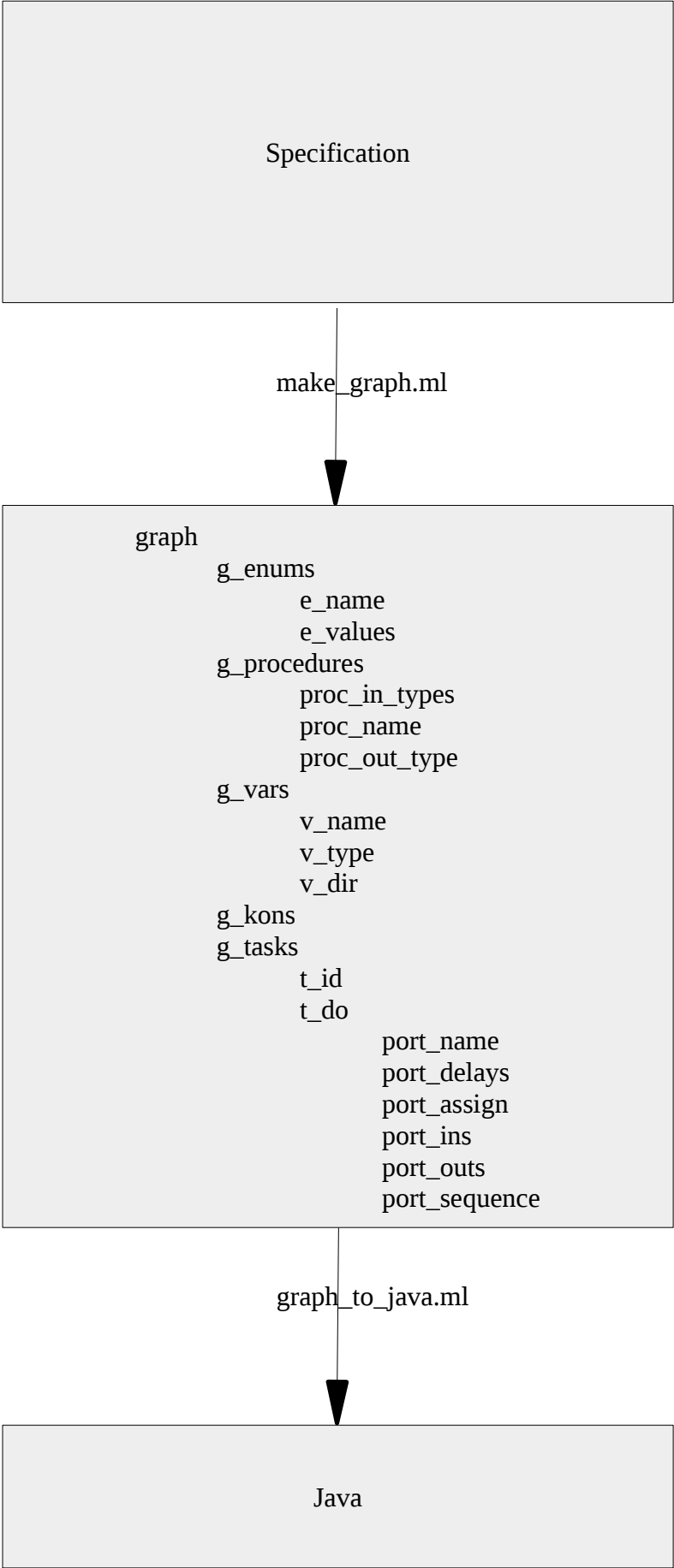
Garantit une spécification valide en vérifiant les différentes contraintes du langage.

Ter no submod

Minimise le programme de sorte à n'avoir plus qu'un seul process, et que la local_process_list soit vide. Ajout de variables locales, assignations, contraintes et instantiations.

Ter arith to call

Convertit les appels aux opérateurs binaires +, -, * par des appels de fonctions.



Template Java :

<p>data</p> <ul style="list-style-type: none">Constantes.javaCount.javaGlobalData.javaSignal.javaSignal_Int.javaSignal_Bool.javaSignal_... .java <p>exceptions</p> <ul style="list-style-type: none">Uncoded_function.java <p>main</p> <ul style="list-style-type: none">Main.java <p>thread</p> <ul style="list-style-type: none">Ter_Runnable.java... .java <p>usable</p> <ul style="list-style-type: none">Usable.java <p><u>Légende :</u></p> <p>Vert fichiers constants</p>	<p>Main :</p> <ul style="list-style-type: none">création globale de GlobalData et de Countcréation de l'exécuteur des tâchescréation des tâches, ajoutées à la liste de GlobalDatadéfinition des dépendances entre tâchestant qu'on peut lire le fichier In<ul style="list-style-type: none">pour chaque tâche n'ayant pas de dépendances<ul style="list-style-type: none">incrémenter Countlancement de la tâcheon attend Countfin de cycle : reset du GlobalDatafin : shutdown de executor <p>Chaque tâche :</p> <p>I - hérite de Ter_Runnable, avec un run qui garantit</p> <ul style="list-style-type: none">1- l'exécution d'une méthode compute et en fonction du résultat<ul style="list-style-type: none">a - la décrémentation du nombre de dépendances des tâches dépendantesb- l'exécution des tâches dont le nombre de dépendance devient null2- la décrémentation de Count <p>II - implémente compute, méthode chargée du calcul spécifique et renvoyant un booléen pour indiquer si oui ou non le calcul s'est bien passé</p> <p>Usable :</p> <ul style="list-style-type: none">1 - Un enum est créé dans Usable pour chaque enum du graph. <p>Pour chaque ajout d'enum, une classe héritant de Signal est créée qui implémente le type de variables Signal<NomEnum></p> <ul style="list-style-type: none">2 - Une fonction est créée pour chaque procedure du graph. <p>Quand on ne connaît pas le code de la fonction, ce dernier consiste en une levée de l'exception Uncoded_function.</p>
--	--