IPHC
Institut Pluridisciplinaire Hubert CURIEN STRASBOURG

WEBINAR SYSTEMC

11 MAI 2010
PAR
Christian WEBER

ARM
cadence
CoWare
DOULOS
FORTE DESIGN SYSTEMS
Mentor Graphics
SYNOPSYS
virtutech
XtremeEDA

LE LANGAGE SYSTEMC

- Objectifs :
  - Présentation d'un langage pour la modélisation, la simulation et la vérification de systèmes Hardware/Software

- Date : 11 MAI 2010

- Présentation : Christian WEBER

- E-mail :christian.weber@iphc.in2p3.fr

IPHC
Institut Pluridisciplinaire Hubert CURIEN STRASBOURG

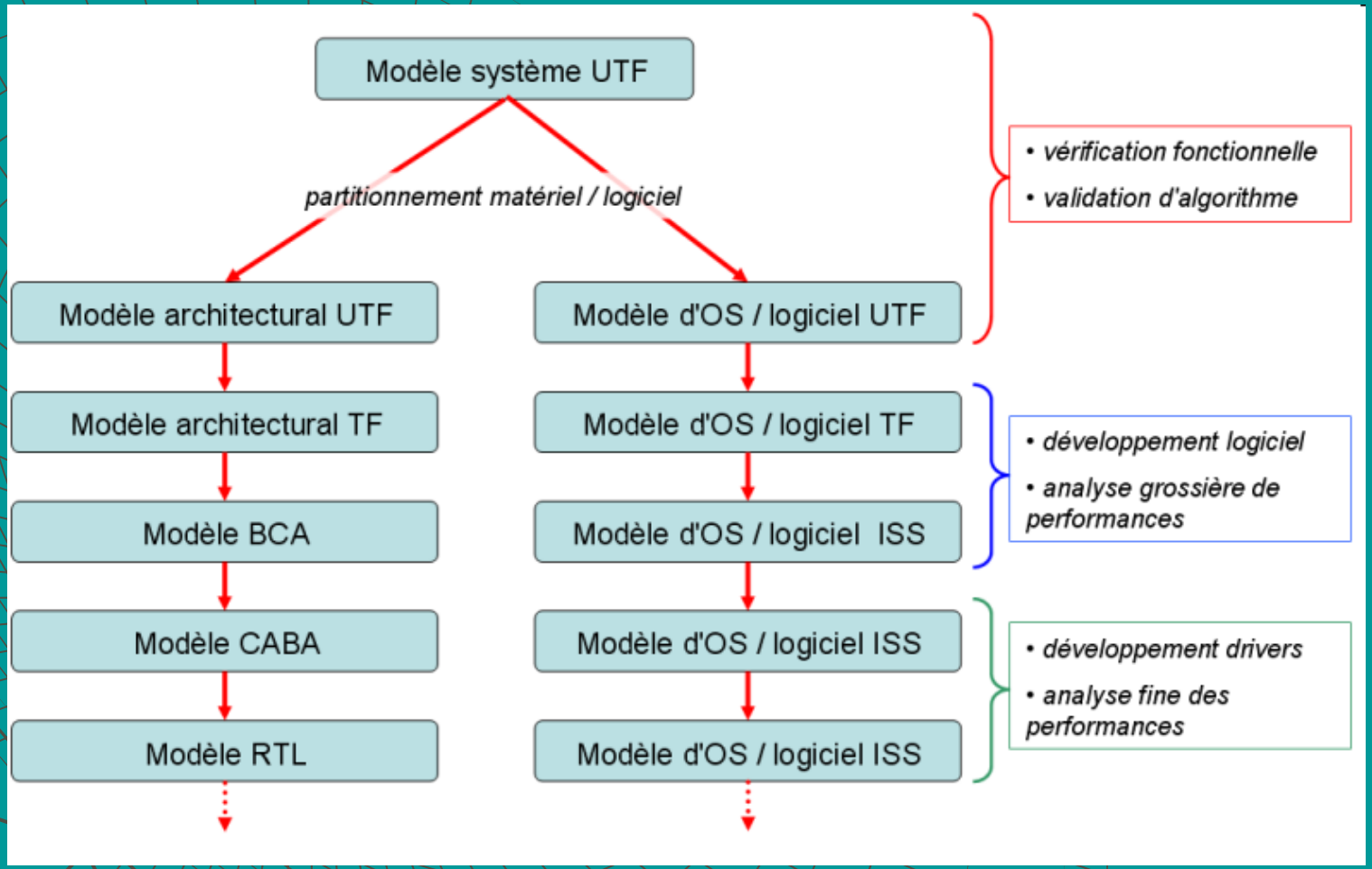
PLAN

- INTRODUCTION
- PARTITIONNEMENT HARDWARE/SOFTWARE
- FLOT DE CONCEPTION SC
- SEMANTIQUE SYSTEMC
- OBJETS SYSTEMC
- MODELE EN COUCHES
- NIVEAUX D'ABSTRACTION SC
- SYSTEMC KERNEL
- SYSTEMC SCHEDULER
- STRATEGIE DE CONCEPTION
- COMPILATION ET RUN-TIME
- ETUDE d'UN EXEMPLE RTL
- CONCLUSION

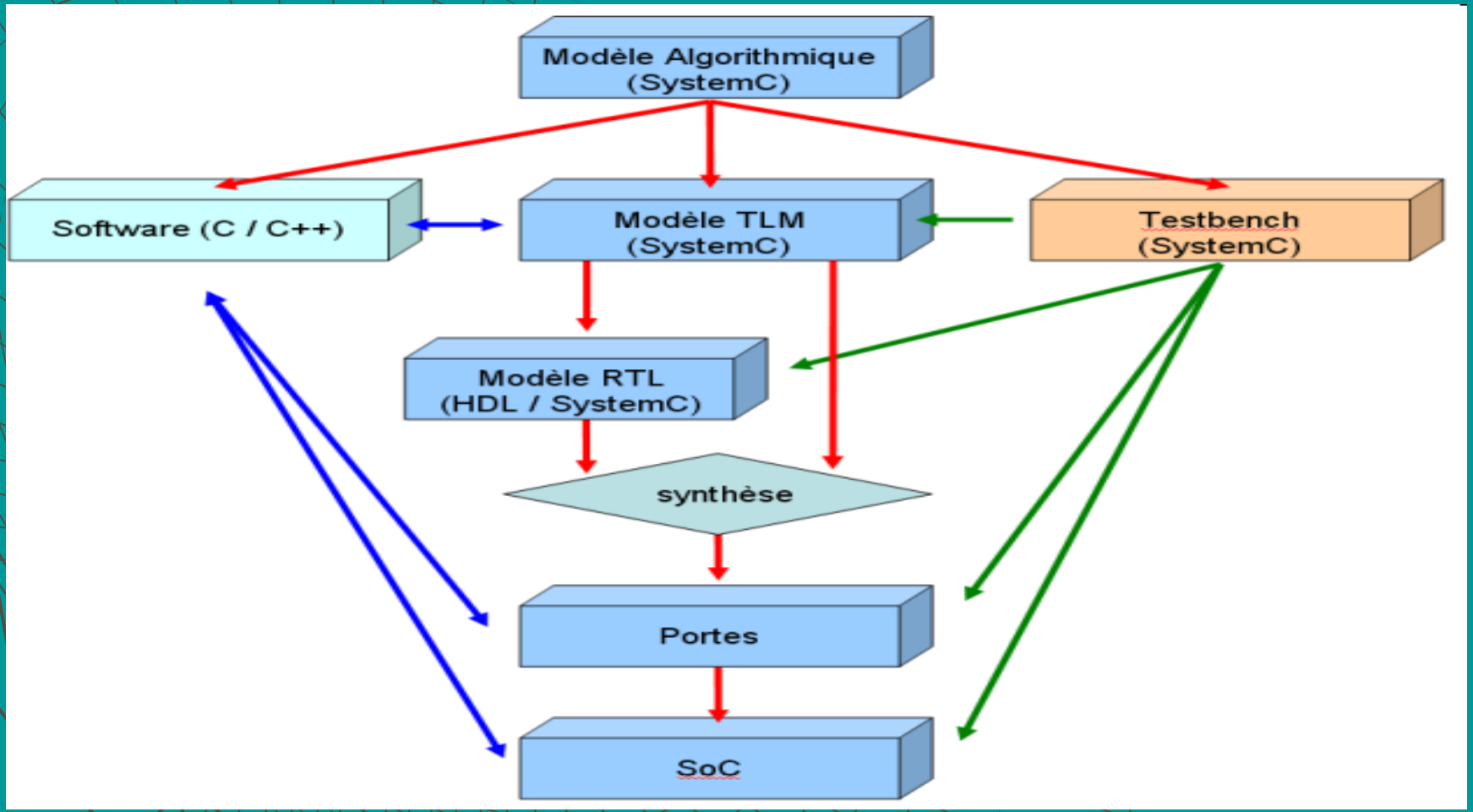
INTRODUCTION

- POURQUOI SYSTEMC ?
- PARTITIONNEMENT HARD/SOFT
- Synthèse et vérification de SoC (System On Chip)
- Permet de se placer à plusieurs niveaux d'abstractions
- Interactions entre spécialistes Hard/Soft(C/C++ ⇔ HDL's)

Partitionnement Hard/Soft



FLOT DE CONCEPTION SC



Sémantique SystemC

- SystemC =>

  Bibliothèques C++ de composants logiciels associés + Noyau de simulation

  =>Basé sur les concepts POO

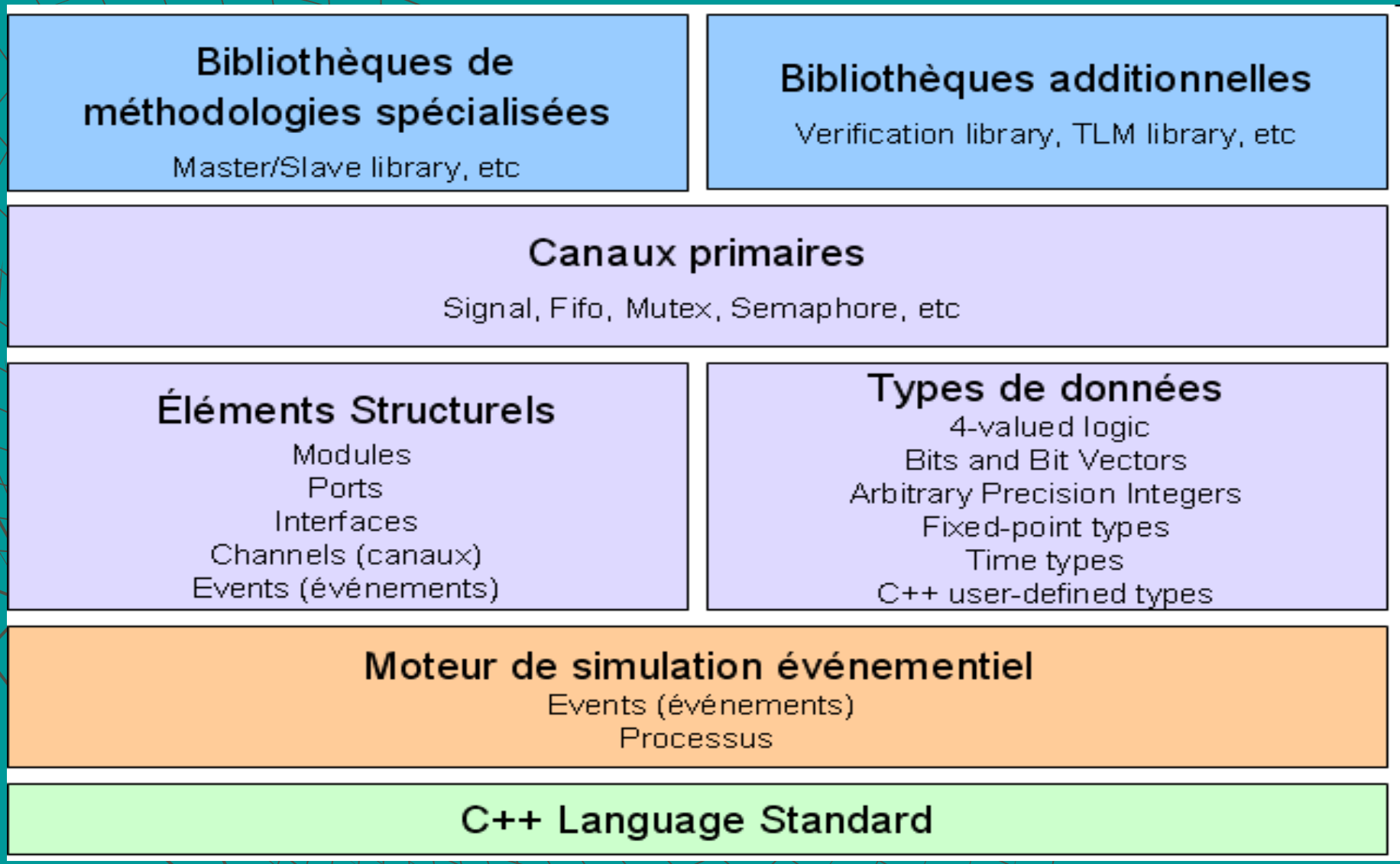
  =>Flexibilité, portabilité…

  =>Programmation générique

  =>« templates » C++

  =>Equivalence pure en C++

OBJETS SYSTEMC



Modèle en couches

REMOTE PROCEDURE CALL (RPC)
SIGNALS, MUTEX, SEMAPHORES, FIFO,…
Channels, Interfaces, Ports
Events & Dynamic sensitivity
SIMULATION KERNEL

Niveaux d'abstraction SystemC

- TLM (Transaction Level Modeling)
- RTL (Register transfert Level Modelling)
- Permet de modeliser de cycles de temps (ou évènementiels purs)

Exploitation de la concurrence dans SystemC

- Les processus concurrents peuvent être représentés par :

  - SC_METHOD()

  - SC_THREADS()

  - SC_CTHREADS()

SystemC Kernel

- Un scheduler qui déclenche l'exécution des tâches dans l'état ready
- Simulations de processus concurrents ;
- Réactif aux évènements synchrones et asynchrones ;
- Les tâches sont gérées de façon quasi-parallèle (coroutines au sens du temps réel) ;
- Pas encore d'outils pour gérer le temps réel (ex. préemption, priorité des tâches…)

SystemC Scheduler

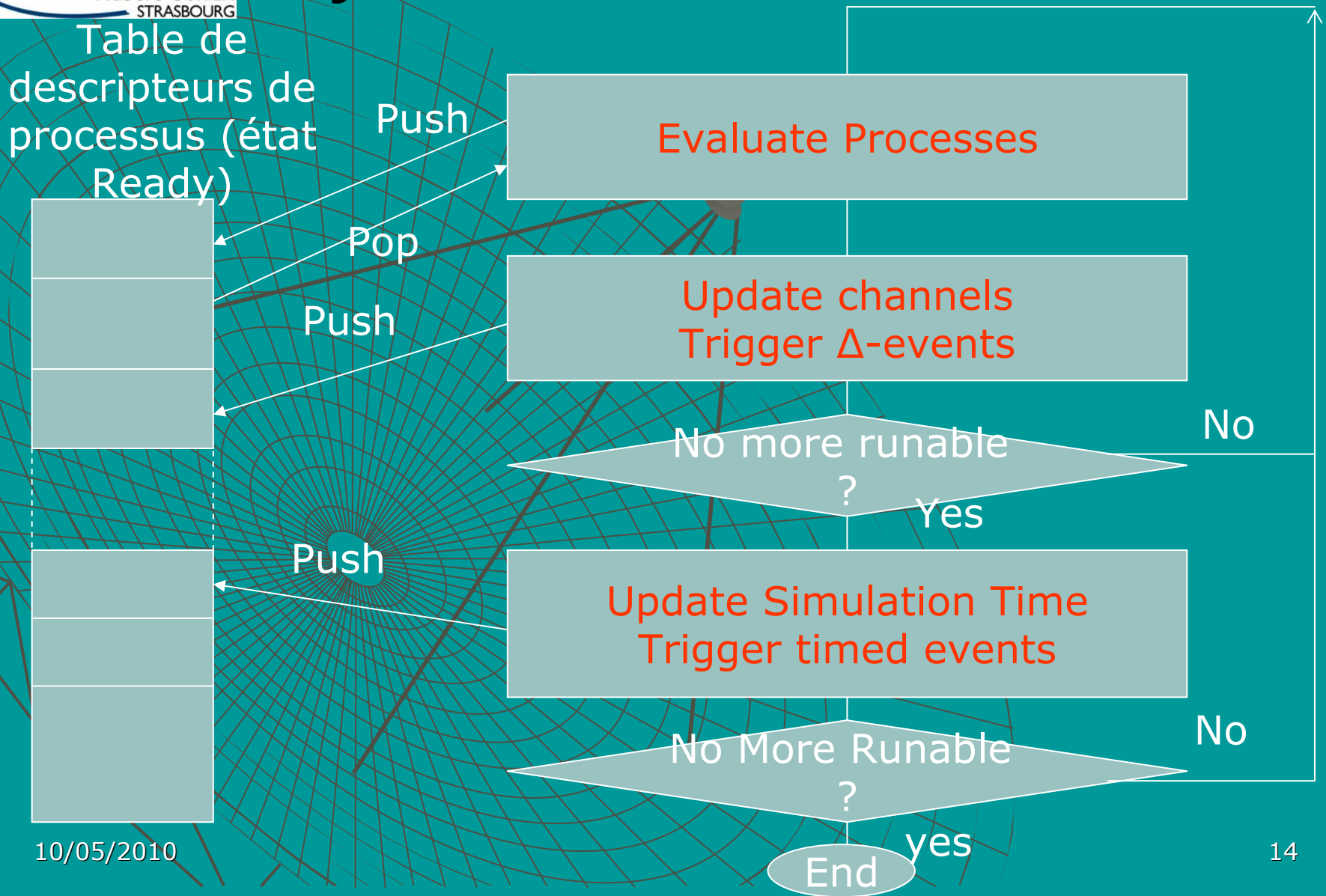
- Fait partie du SystemC Kernel

  -Déclenche l'exécution des processus dans l'état « Ready »

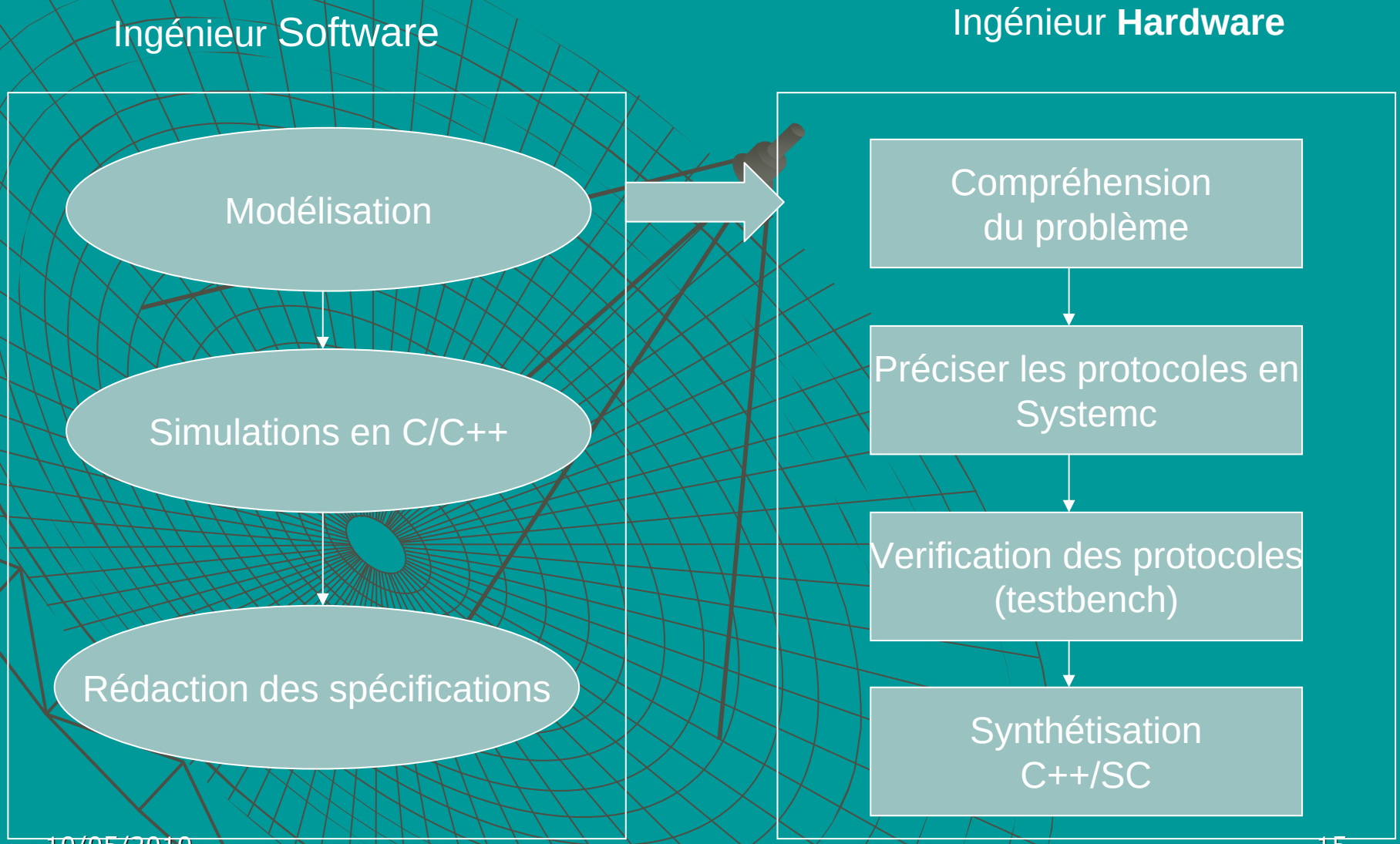
  -Met à jour les ports E/S gérés par ces processus ;

  =>Cette phase constitue un cycle DELTA (garanti le DETERMINISME entre les phases)

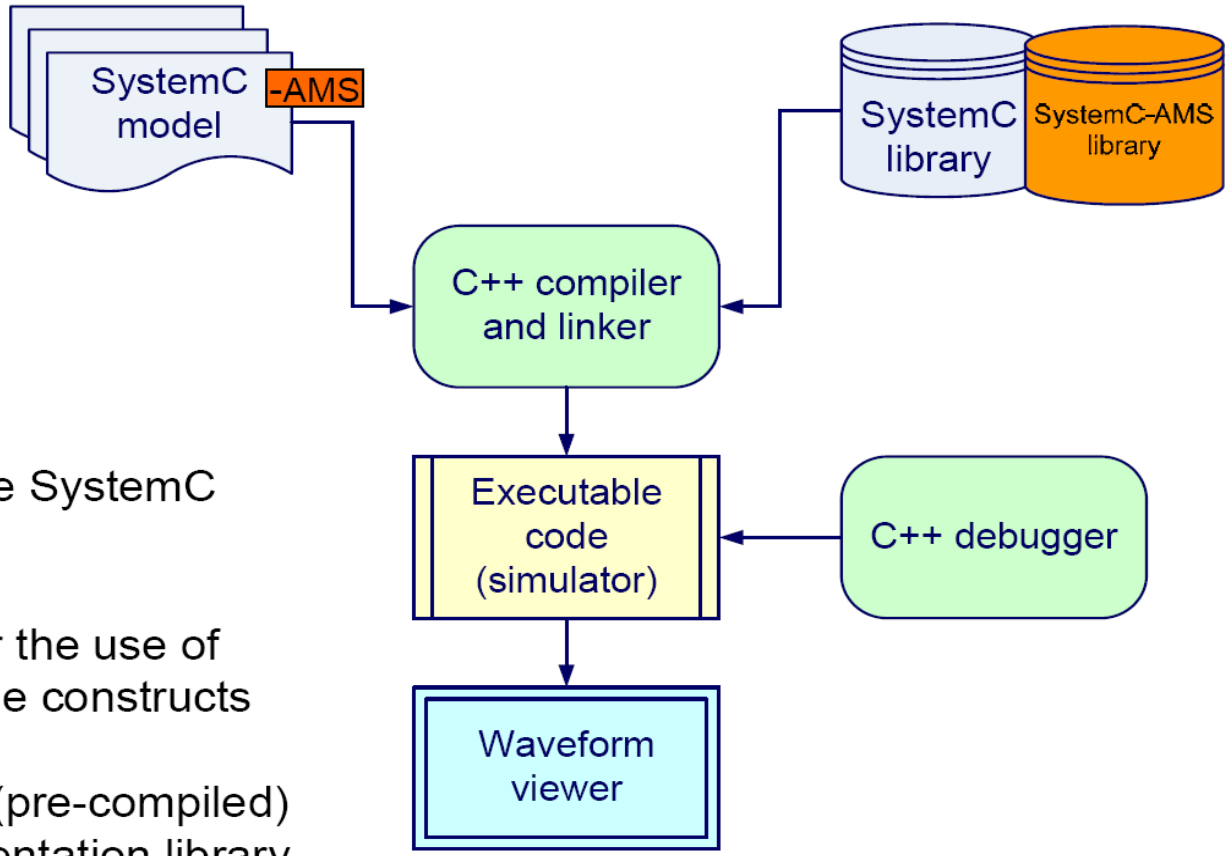
SystemC Scheduler



tégie de conception Systemc



Compilation et run-time



- No changes to the SystemC language
- No restrictions for the use of SystemC language constructs
- Use of the same (pre-compiled) SystemC implementation library

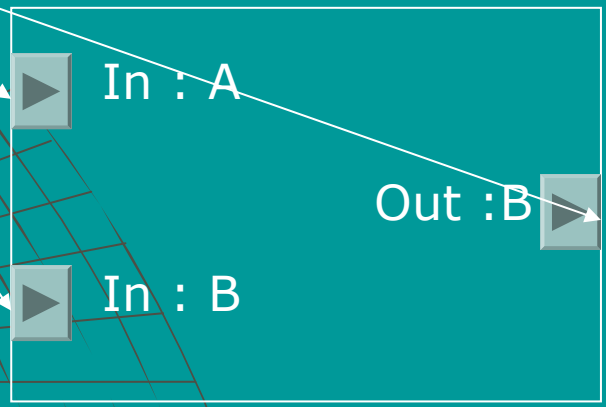
Etude d'un exemple simple (au niveau RTL)

Header

```
//Nand2.h
#include <systemc.h>

SC_MODULE(nand2) {
    // ports definitions

    sc_in<sc_bit> A;
    sc_in<sc_bit> B;
    sc_out<sc_bit> F;

    void do_nand2();

    SC_CTOR(nand2)
    {

        SC_METHOD(do_nand2)
        sensitive << A << B;

    }

};
```

Ports I/O

Nom_module : nand2

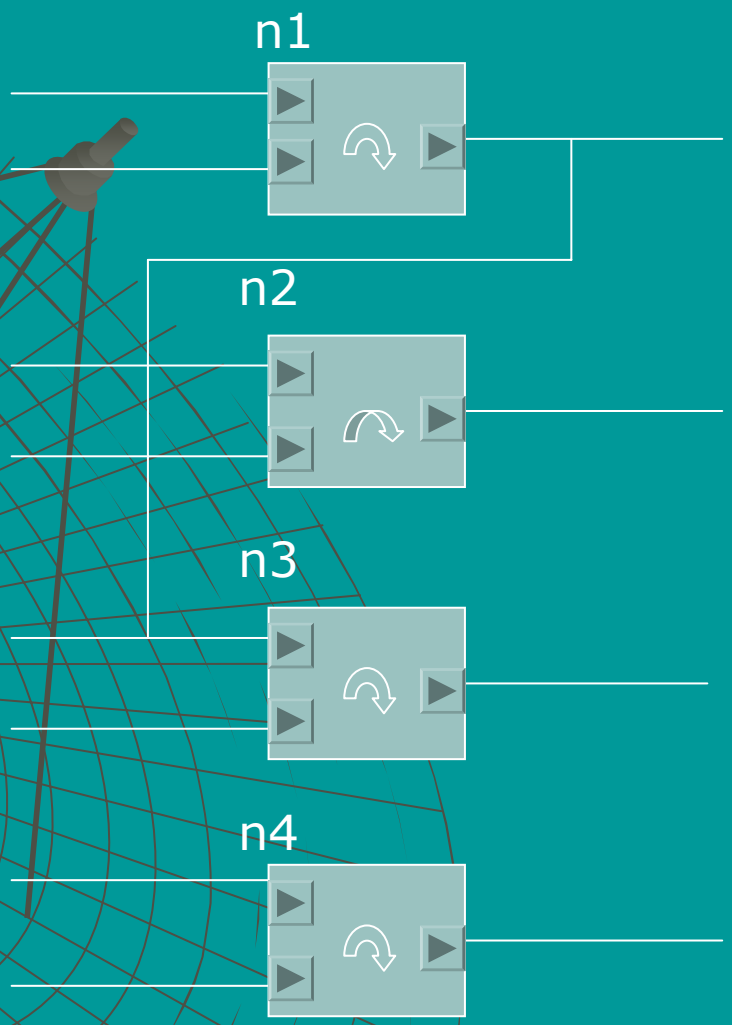


Constructeur

Déclaration de do_nand2()
Dans la table de descripteur de processus
Du noyau

Etude d'un exemple RTL

```
// définition des signaux
// un signal est tjrs relie a 2 ports

sc_signal<sc_bit> S1;
sc_signal<sc_bit> S2;
Sc_signal<sc_bit> S3;

    SC_CTOR (exor2)
    {
            n1.A(A);          n3.A(S1);
                              n3.B(B);
                              n3.F(S3);

            n1.B(B)
            n1.F(S1);

            n2.A(A);
            n2.B(S1);
            n2.F(S2);

            n4 << S3 << S3 << F ;
```



SIMULATIONS MIXTES COUPLAGES

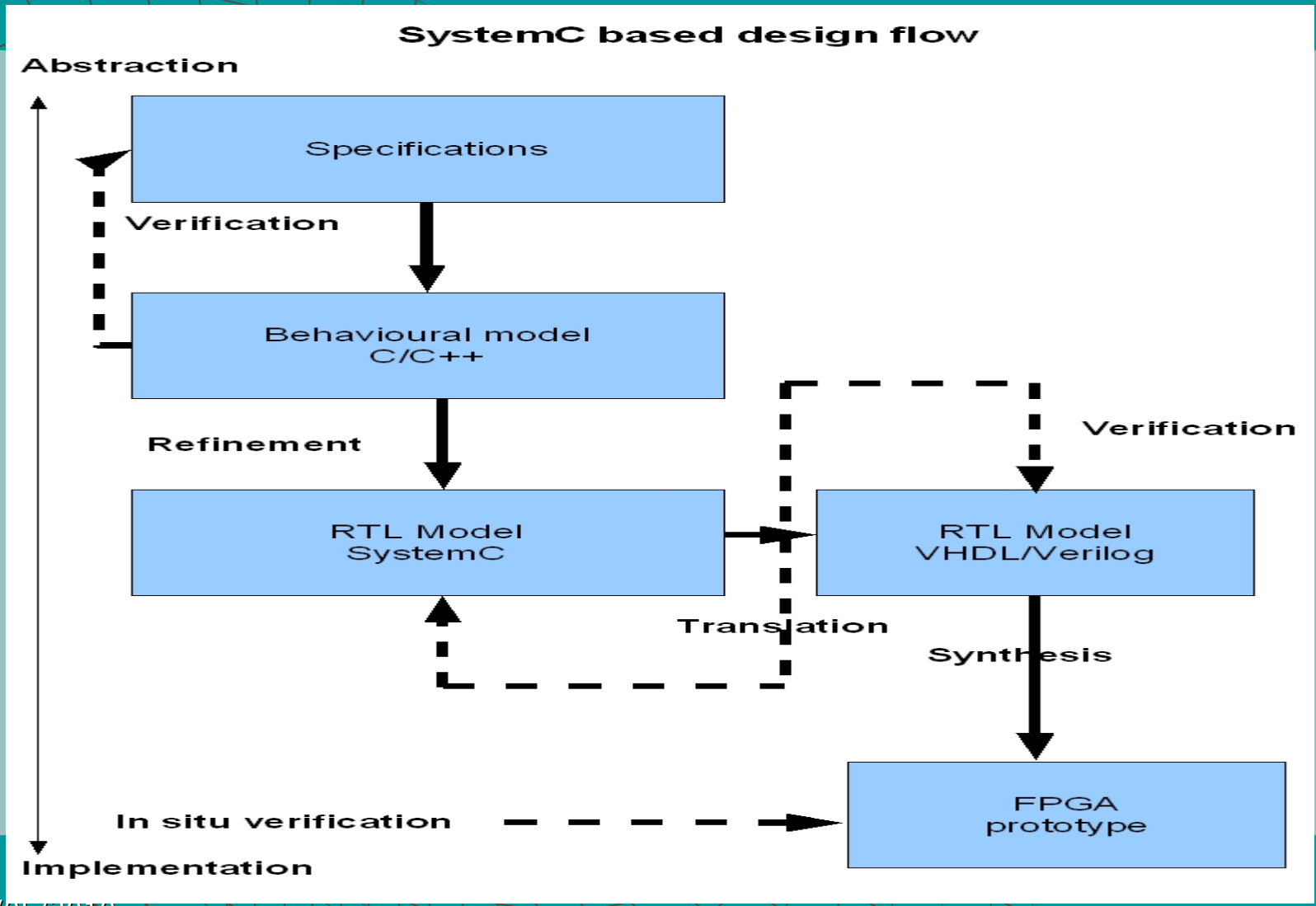
- Possibilités d'interfaçage avec d'autres environnement :

  - Ptolemy II (Actors…);
  - Matlab Simulink;
  - VHDL

SystemC Distribué

- Parallélisation de Noyaux SC dans un environnement distribué (ex. Cluster de PC, grille de calcul…) ;
- Synchronisation des Noyaux SC ;
- Communication par l'intermédiaire de sockets (Berkeley Sockets)

SYNTHESE DIRECTE



Exemple : FPGA Synthesis

La synthèse directe est réalisée par des outils commerciaux :

- Mentor Graphics, Synopsys

La synthèse génère un fichier au format EDIF :

=> génération du code VHDL;

=> Vérification par une API proposé par le constructeur du circuit (ex : XILINX, LATTICE,…)

Concepts Hardware

Les systèmes Hardware sont des processus parallèles (concurrents et/ou séquentiels) :

=>Ces processus sont activés sur des changement d'états de signaux

=>Les systèmes transfèrent des données synchronisées sur des horloges

Systemc pour la synthèse de circuits

- Circuits logiques synchrones ET asynchrones (clockless)
- A terme possibilité de synthétiser des circuits sans passer par une étape RTL
- Environnement de développement :

  Mentor Graphics, Cadence, Synopsys…

CONCLUSION

- C/C++ plus de souplesse par rapport aux langages HDL STANDARTS (VUE UNIFIEE DES ETAPES D'UN PROJET)
- Tendance actuelle : synthèse directe de circuits en C/C++ !!
- Certains compilateurs peuvent synthétiser(sans passer par VHDL ou VERILOG)

Conclusion(con't)

- Systemc évolue rapidement :
  - Simulations analogiques (SC-AMS)
  - Simulations mixtes
  - Outils associés en cours de développement
  - Dans les versions récentes :

    Autres outils proposés :

    - processus dynamiques ;
    - callbacks, sc_export ;
    - debugger
    - correction de bugs

SITES A CONSULTER

- http:www.forteds.com
- Propose une introduction à Systemc en ligne

- Autres sites intéressants :

  OSCI (Open Source SystemC Initiative)

  http://www.systemc.org

  www-ti.informatik.uni-tuebingen.de

Merci pour votre attention!!

- QUESTIONS ??????