

SÉANCE 6

**LANGAGE DE
SPÉCIFICATION ET DE
DESCRIPTION(SDL)**



uOttawa

L'Université canadienne
Canada's university

SUJETS

Les objectifs de SDL

Structure de SDL

- Système
- Bloc

Comportement de SDL

- Processus

Signaux de SDL

- Adressage de processus implicite et explicite

Exemple de définition de système SDL



uOttawa

L'Université canadienne
Canada's university

QU'EST-CE QUE LE SDL

SDL est une langue formelle développée et standardisée par le ITU-T

Elle est prévue pour la spécification d'applications complexes, événementielles, à temps réel, et interactives

Ces applications impliquent typiquement plusieurs activités concurrentes qui communiquent en utilisant des signaux discrets



uOttawa

L'Université canadienne
Canada's university

QU'EST-CE QUE LE SDL

SDL décrit l'*architecture*, le comportement et les données de systèmes distribués dans des environnements à temps réel

Il est utile pour spécifier la spécification du comportement du système de télécommunication

SDL couvre différents niveaux d'abstraction, depuis une vue d'ensemble étendue jusqu'au niveau de conception détaillé



uOttawa

L'Université canadienne
Canada's university

DOMAINE D'APPLICATION

Type de systèmes:

- Temps réel
- Interactif
- Distribué
- Hétérogène

Type d'information:

- Comportement, structure

Niveau d'abstraction:

- Vue d'ensemble jusqu'aux détails



uOttawa

L'Université canadienne
Canada's university

UTILISATION DU SDL

SDL peut être utilisé pour des étapes variées du développement:

- Exigence formelle de modélisation
- Conception
- *Implémentation?*
 - *SDL n'était pas originalement prévu pour l'implémentation, mais la traduction au code est possible*



uOttawa

L'Université canadienne
Canada's university

OBJECTIFS

L'utilisation d'outils basé sur les ordinateurs afin de créer, maintenir, et analyser les spécifications

- Descriptions formelles de haute qualité produites d'une façon améliorée, rapide et moins chère
- Analyse de spécifications pour la complétion et la justesse
- Support de la communication humaine et compréhension des exigences

Peut être utilisée pour l'optimisation du concept

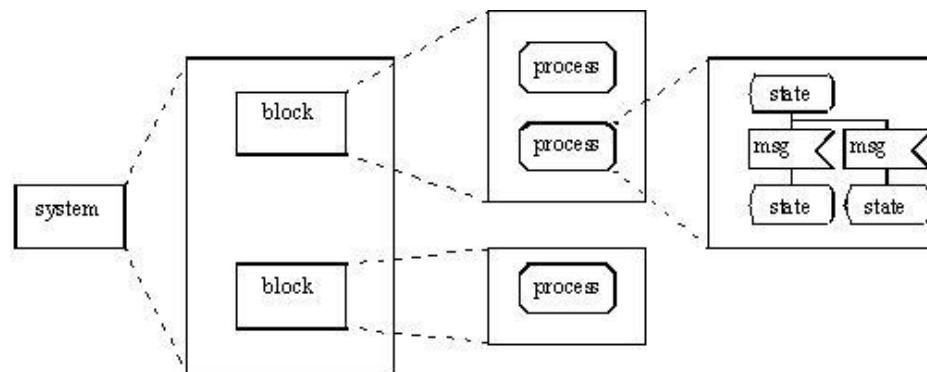
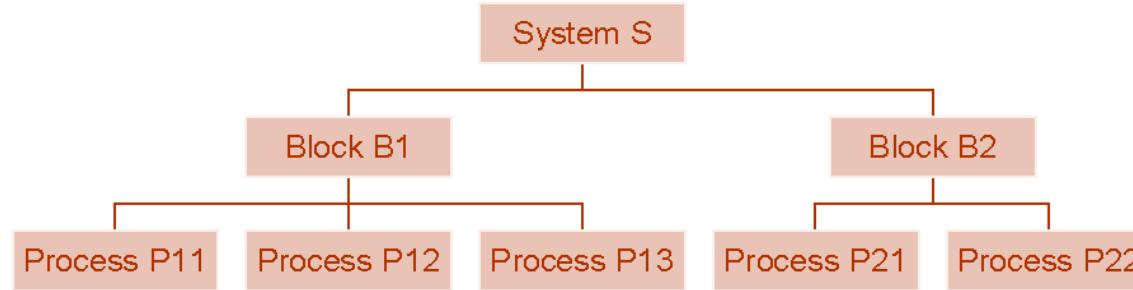
- Fournit les programmeurs d'un moyen facile pour effectuer les vérifications, les validations du concept

Spécification facile de la vérification de conformité d'implémentation

STRUCTURE SDL

Comprend trois niveaux principaux hiérarchiques:

- Système
- Bloc
- Processus

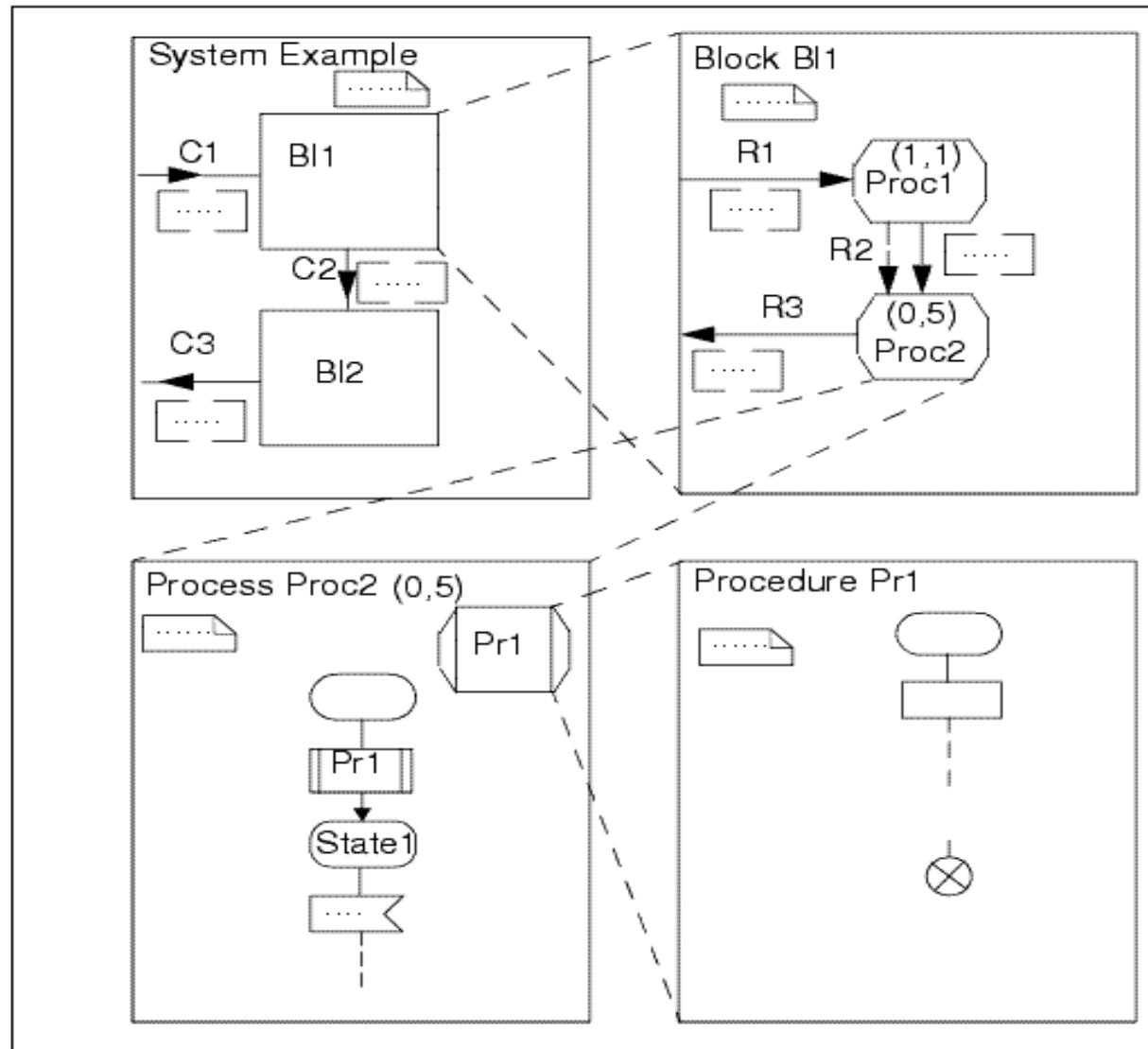




uOttawa

L'Université canadienne
Canada's university

STRUCTURE SDL





uOttawa

L'Université canadienne
Canada's university

STRUCTURE SDL

Un système contient un ou plusieurs blocs, interconnectés ensemble et avec les limites du système par des canaux

- Les processus communiquent ensemble en utilisant des signaux envoyés par des canaux

Le bloc est le concept structural principal

Un bloc peut être réparti en sous-blocs

Un canal est un moyen de transport des signaux

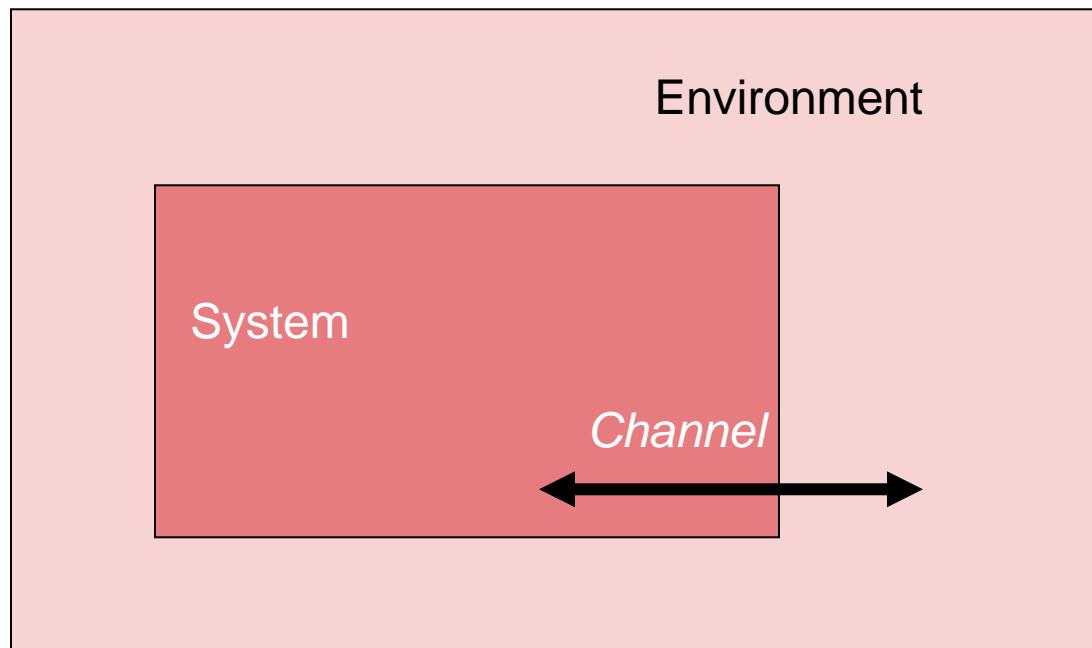


uOttawa

L'Université canadienne
Canada's university

SYSTÈME

Le système représente une machine abstraite qui communique avec son environnement



SYSTÈME

System name

System Deamongame

1(1)

/* This system is a game having any number of playes. The players belong to the environment of the system. A "Deamon" in the environment of the system sends Bump signals randomly to the system. A player has to guess whether the number of the Bump signals received by the system is odd or even. The guess is made by sending a Probe signal to the system. The system relies by the signal Win if the number received Bump signals is odd, otherwise by the signal Lose.

The system keeps track of the score of each player. Aplayer can ask for the current value of his score by the signal Result, which is answered by the system with the signal score.

Before a player can start playing, he must log in. This is accomplished by the signal Newgame . A player logs out by the signal Endgame. */

SIGNAL Newgame, Probe, Result, Endgame, Gameid, Win, Lose,
Score (Integer), Bump;

Text symbol

Signal declaration

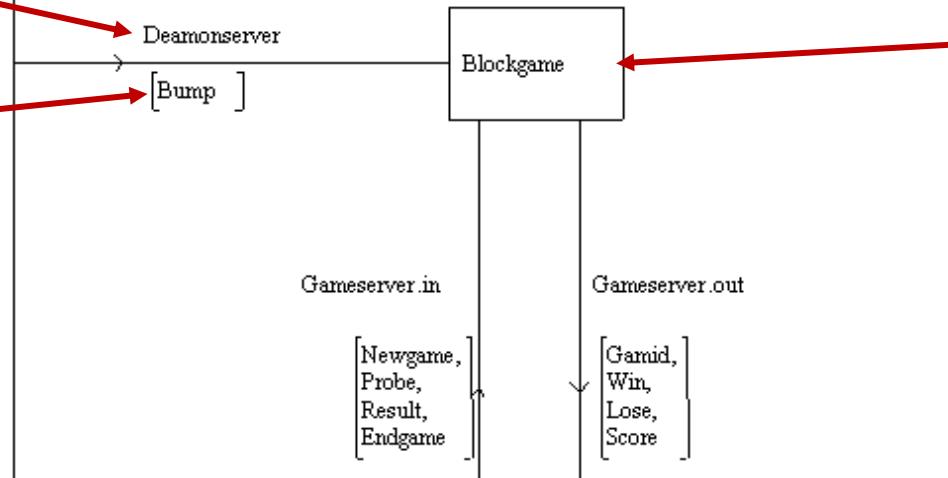
Channel

Deamonserver



Block

Signal

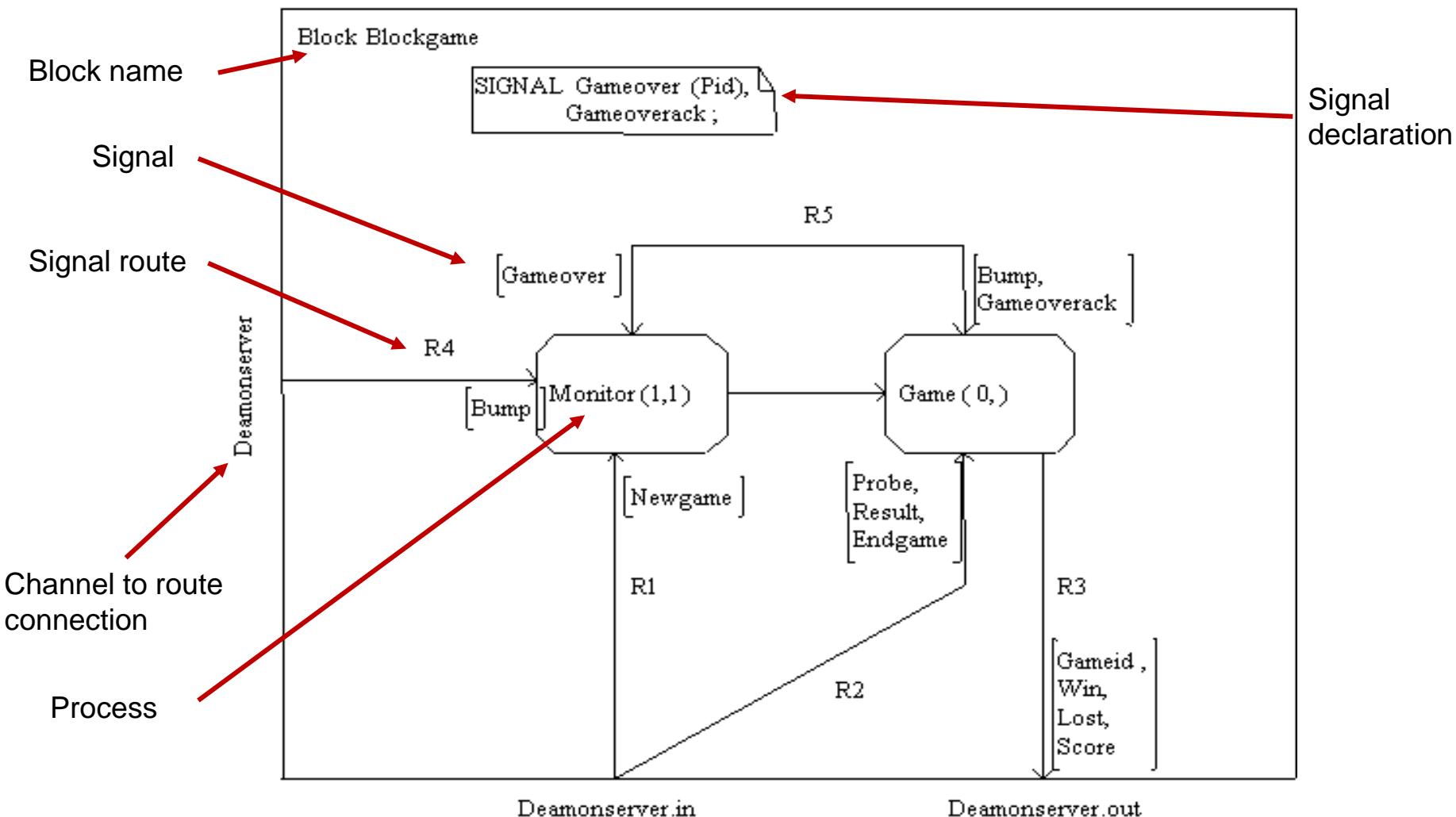


BLOC



uOttawa

L'Université canadienne
Canada's university





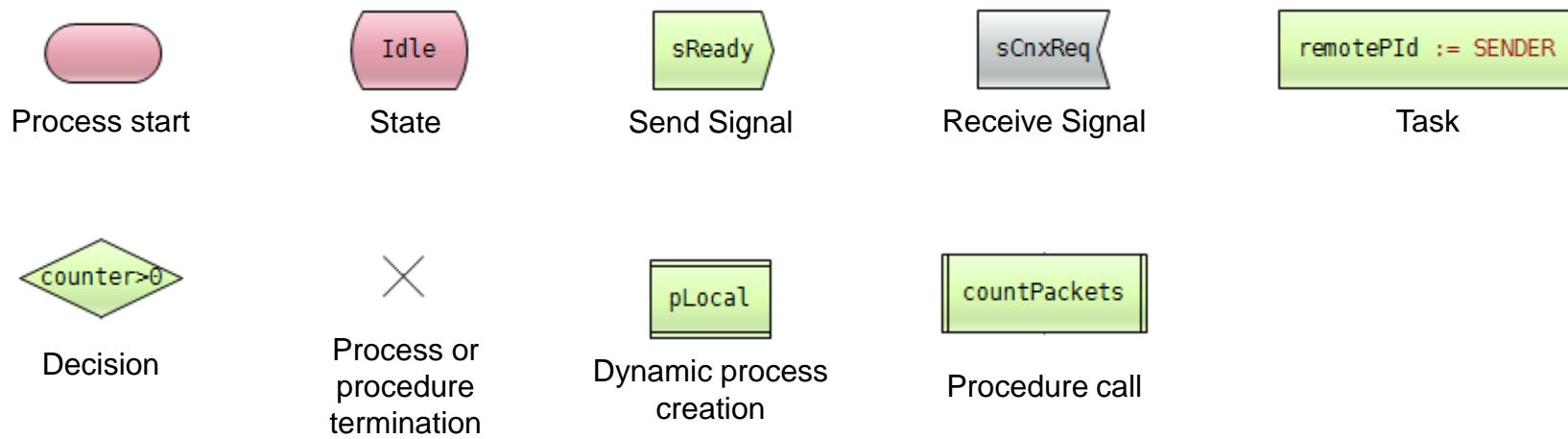
PROCESSUS

Un processus dans SDL est une machine d'état finie étendue

Le comportement d'une machine d'état finie est décrite par les états et les transitions

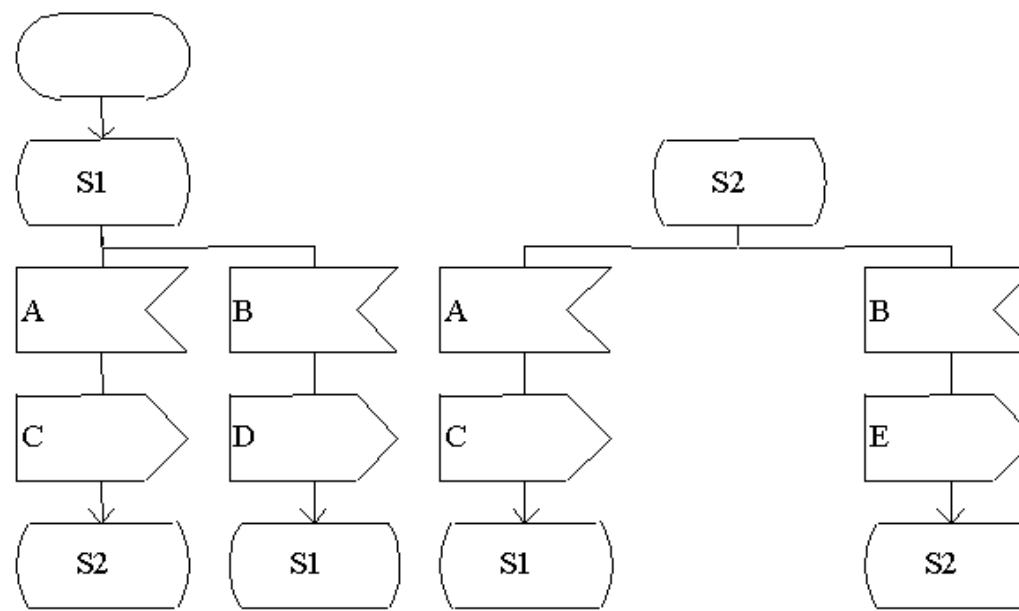
Une description de processus est livrée à travers un diagramme de processus

Ci-dessous sont les éléments de base utilisées pour la description d'un processus:



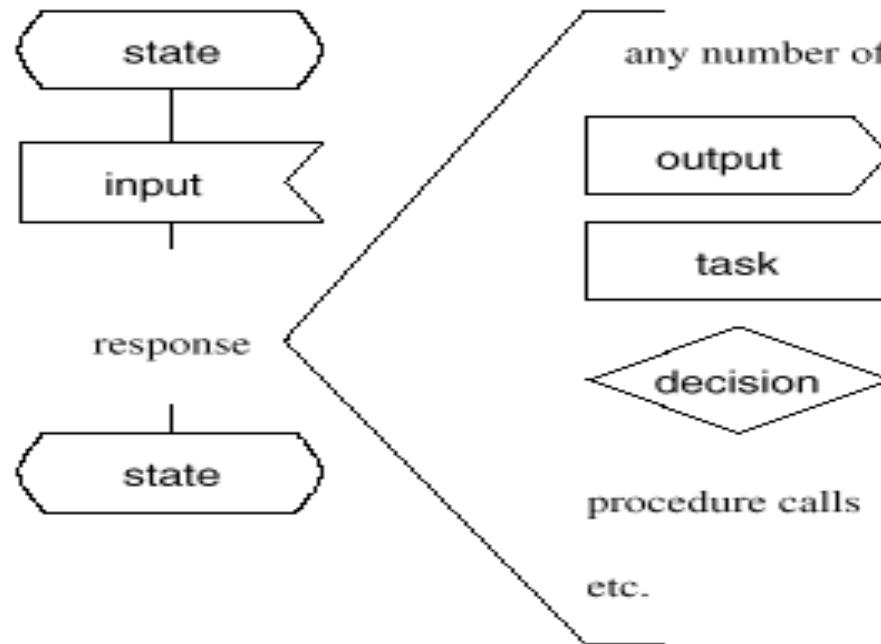
PROCESSUS

Process Example



FORMULE DU DIAGRAMME DE PROCESSUS

En général, chaque transition d'état possède la formule suivante



Dans une transition d'état, un processus reste dans son état actuel jusqu'à ce qu'un événement de données prévu est reçu.



uOttawa

L'Université canadienne
Canada's university

PROCÉDURE

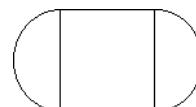
La construction de procédure dans SDL est similaire à celle connue à partir des langues de programmation

Une procédure est une machine d'état finie en dedans d'un processus

- Elle est créée lorsqu'un appel de procédure est interprété, et meurt lorsqu'il est terminé

Une description de procédure est similaire à une description de processus, avec quelques exceptions.

Le symbole de départ est remplacé par le symbole de départ de la procédure



DÉcrire le comportement avec SDL

Le comportement d'un système est constitué par la combinaison de comportements des processus dans le système

Un processus est défini comme étant une machine d'état finie, qui travaille d'une façon autonome et concurrentielle avec d'autres processus

Un processus réagit aux stimuli externes en accord avec sa description

Un processus est soit dans un état d'attente, ou il effectue une transition entre les deux états



uOttawa

L'Université canadienne
Canada's university

DÉcrire le comportement avec SDL

La coopération entre les processus est effectuée d'une façon asynchrone à l'aide de messages discrets appelés signaux

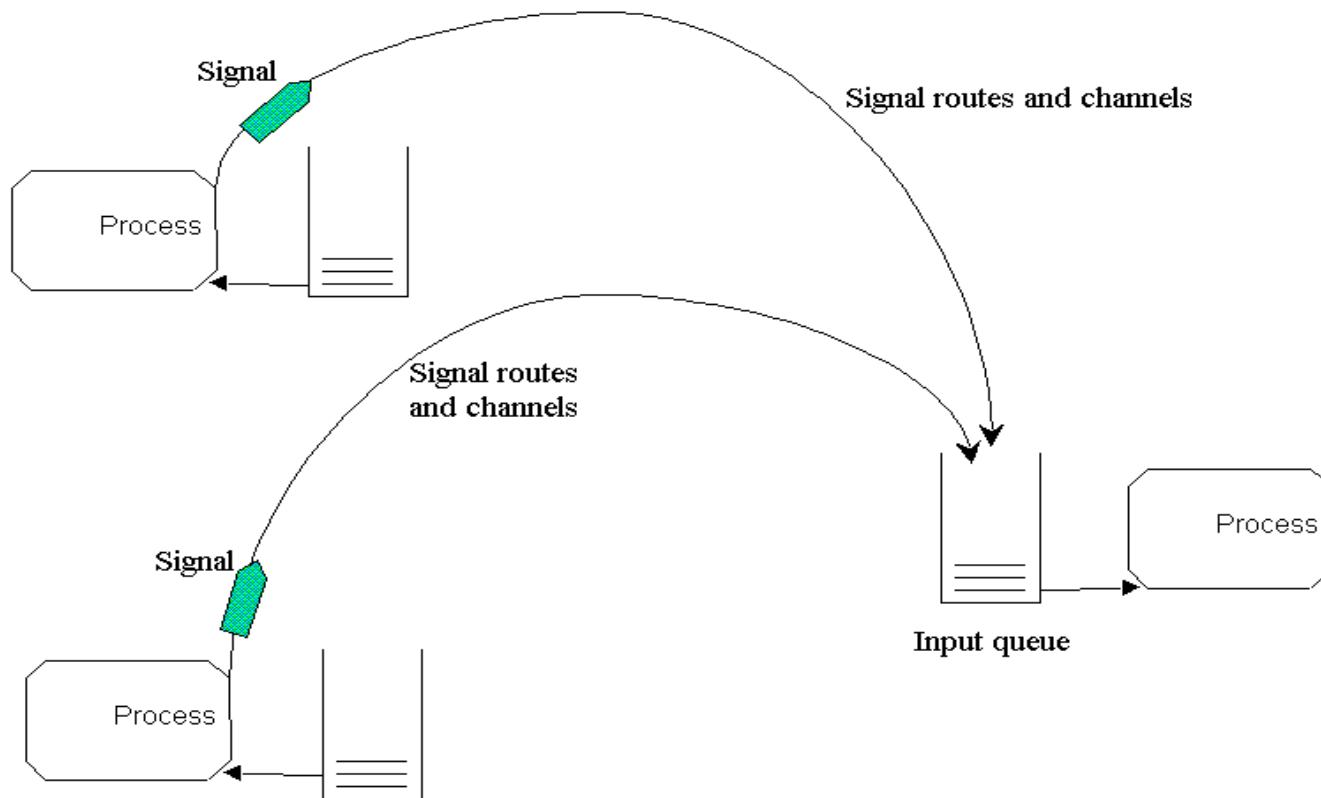
Chaque processus possède une file infinie de données associée, qui agit comme une file FIFO

Tout signal qui arrive au processus rentre dans sa file d'attente de données

- L'ensemble de signaux de données prédéfinis est défini comme étant les signaux que le processus se prépare à accepter

DÉcrire le comportement avec SDL

Lorsqu'un signal a initié une transition, il est éliminé de la file d'attente de données





uOttawa

L'Université canadienne
Canada's university

VARIABLES DU PROCESSUS

Un processus peut utiliser et manipuler localement les données stockées dans les variables

- Celles-ci sont définies, dans un symbole de texte ajouté à la définition du processus
- Le mot-clé DCL introduit la définition d'une ou plusieurs variables dans une boîte de texte
- DCL représente Digital Command Language (langue digitale de commande)

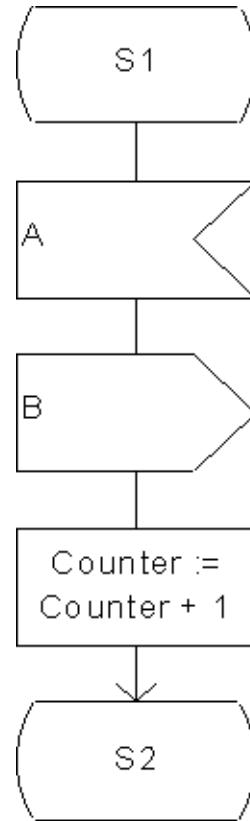
```
DCL
Counter Integer := 0;
```

Lorsqu'il y a plus d'une variable définie après DCL, elles sont séparées par des virgules

- La liste de définition se termine par un point-virgule

VARIABLES DU PROCESSUS

Durant une transition, le processus peut utiliser et manipuler ses propres variables locales, en utilisant l'élément de tâches





uOttawa

L'Université canadienne
Canada's university

IDENTIFICATEUR DE PROCESSUS

Chaque instance de processus a un identificateur unique

L' identificateur n'est pas déterminée par le modélisateur

- Elle est créée par la machine SDL (i.e. programme utilisé pour la simulation) durant la création du processus
- La machine SDL garantit que chaque instance de processus reçoit un identificateur unique

L' identificateur d'une instance de processus n'est pas identique au nom du processus

- Il existe plusieurs instances de processus à partir d'une seule définition de processus



uOttawa

L'Université canadienne
Canada's university

IDENTIFICATEUR DE PROCESSUS

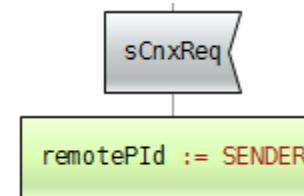
Afin de faire référence à un instance de processus, vous pouvez utiliser les mots-clés suivants

- SELF: retourne l' identificateur de l'instance de processus lui-même
- SENDER: retourne l'identificateur de l'instance de processus duquel le dernier signal consommé a été envoyé
- OFFSPRING: retourne l' identificateur de l'instance de processus qui a récemment été créé par ce processus
- PARENT: retourne l'adresse du processus créateur

L' identificateur de l'instance de processus peut être assignée à un variable de type PID

```
DCL
remotePID PID := NULL,
```

Déclare un variable de type PID



Affecte SENDER à la variable PID déclaré précédemment (sauvegarde pour faire référence dans le future)



uOttawa

L'Université canadienne
Canada's university

ADRESSAGE DU SIGNAL

Pour tout signal envoyé par un processus, il doit y avoir une seule destination

La destination peut être spécifiée:

- Implicitement
- Explicitement



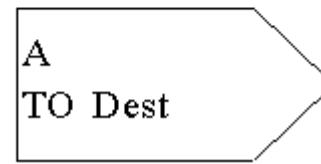
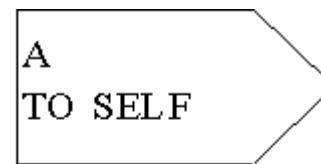
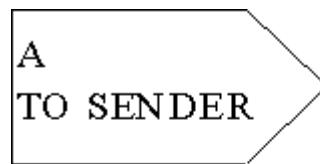
uOttawa

L'Université canadienne
Canada's university

ADRESSAGE EXPLICITE

Le SDL définit le mot clé « TO » pour l'adressage explicite des processus

Le mot-clé TO est utilisé dans une donnée, et il est suivi par une expression qui contient l'identificateur de la destination du processus



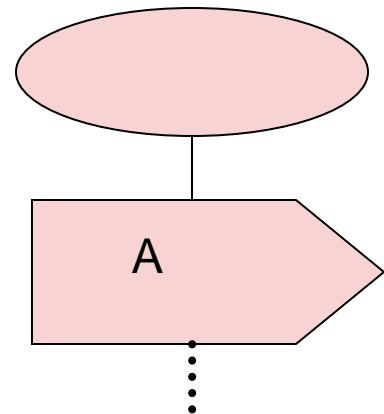
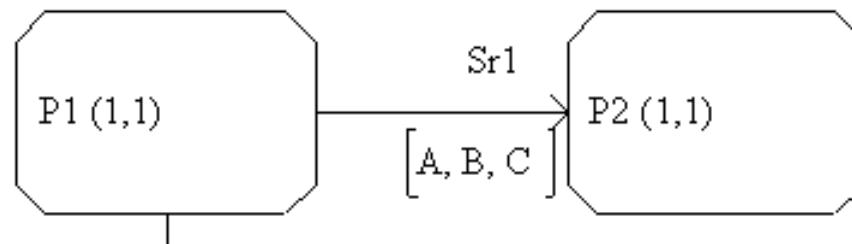


uOttawa

L'Université canadienne
Canada's university

ADRESSAGE IMPLICITE

La spécification explicite d'une adresse de spécification n'est pas nécessaire si la destination est uniquement définie par la structure du système

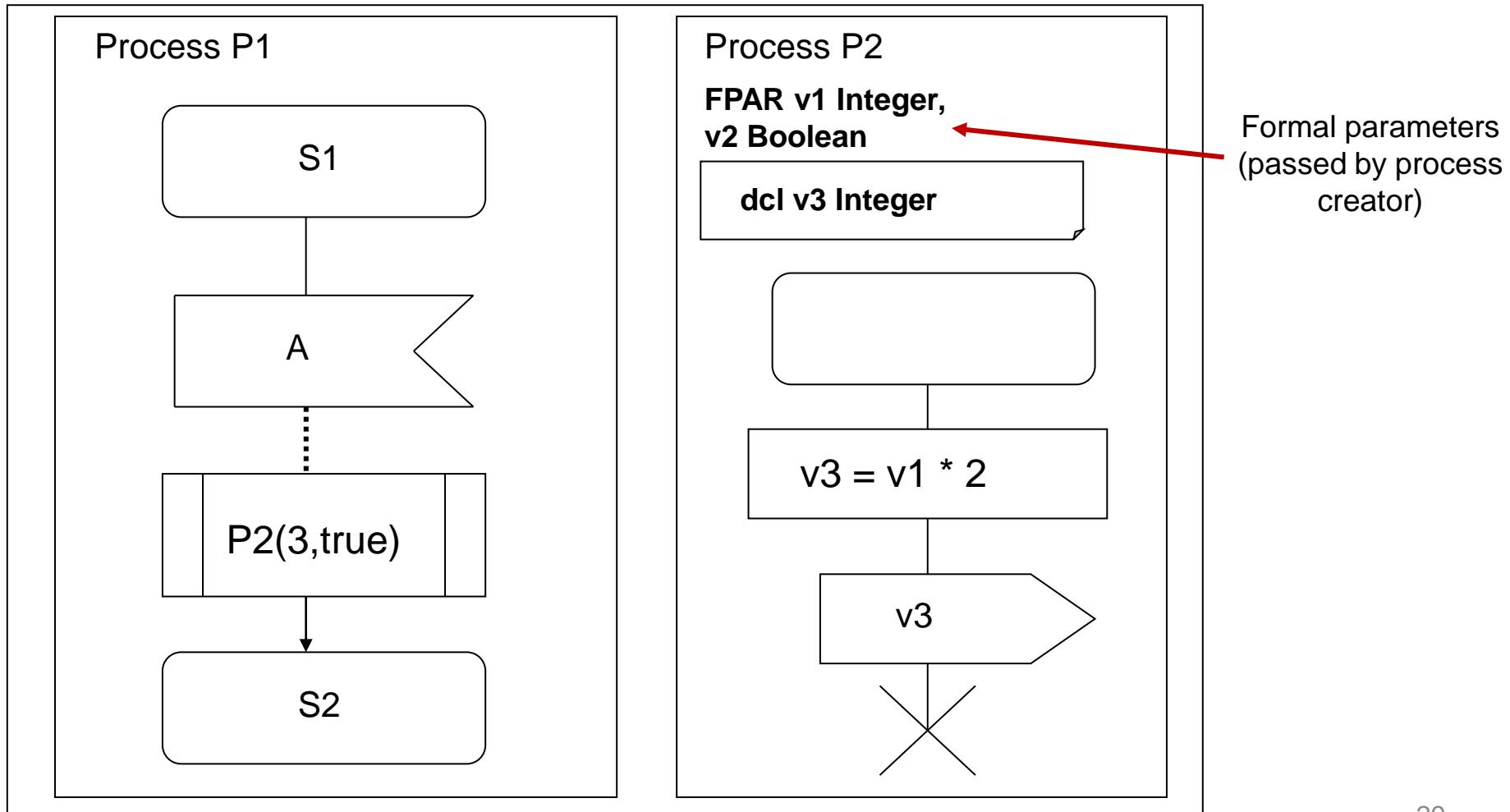


CRÉATION DE PROCESSUS/ TERMINAISON

Les processus peuvent être créés par d'autres processus dynamiquement au temps de l'interprétation

Les processus créateur et créé doivent appartenir au même bloc

CRÉATION DE PROCESSUS/ TERMINAISON





uOttawa

L'Université canadienne
Canada's university

EXAMPLE DE SMART HOME (MAISON INTELLIGENTE)

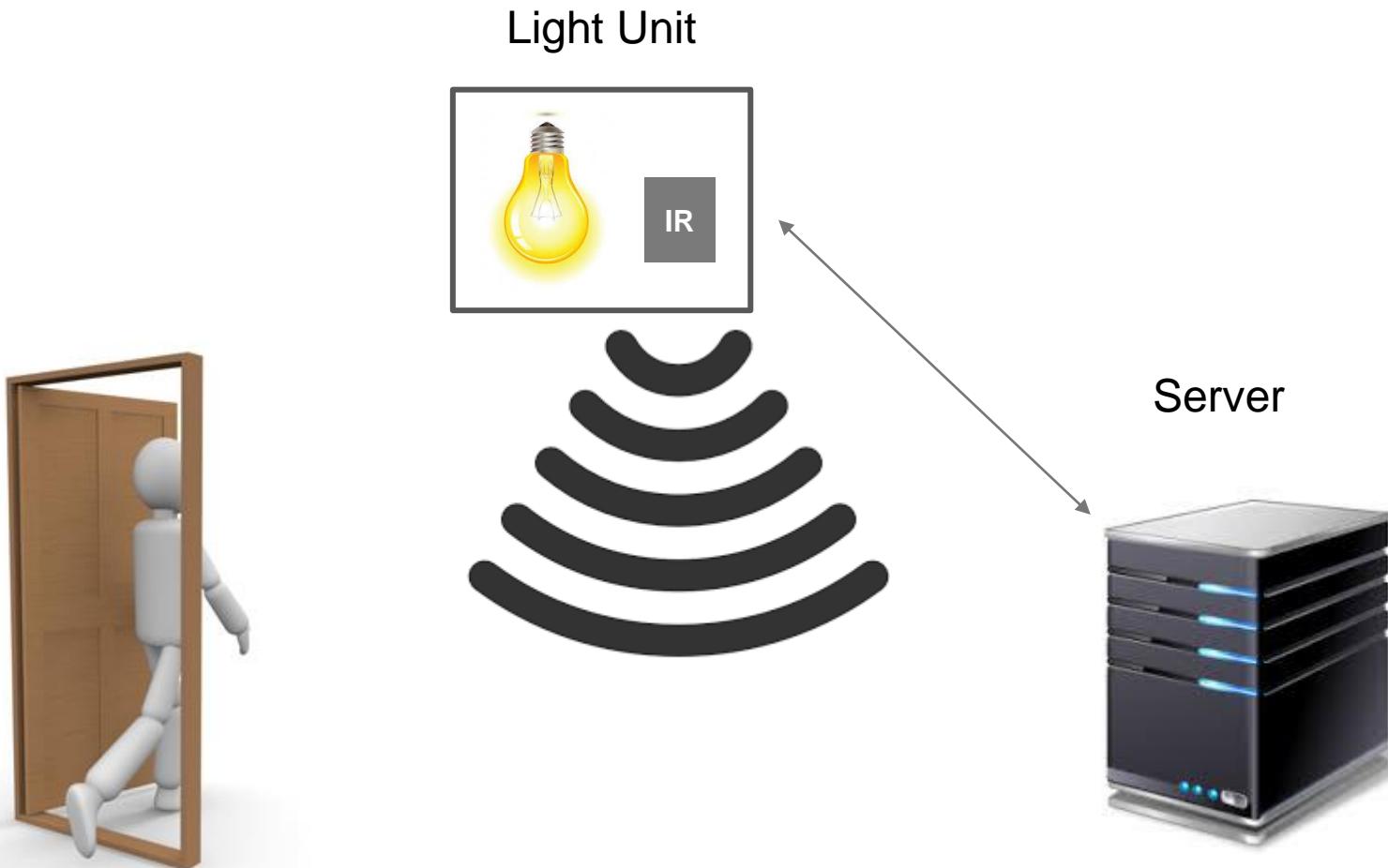
Nous allons modéliser un petit module de système de maison intelligente

Dans une maison intelligente, un processus central qui exécute sur un serveur contrôle plusieurs objets intelligents en dedans de la maison

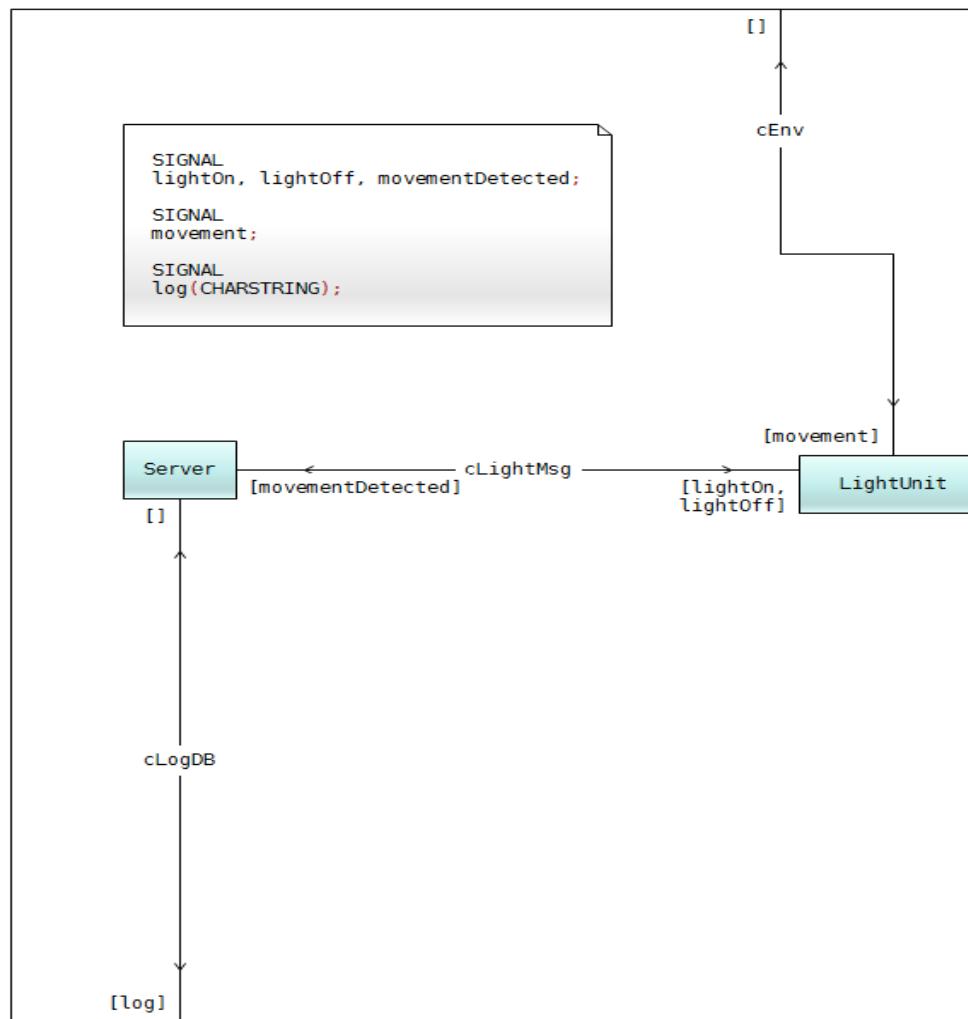
Nous allons modéliser l'interaction entre le processus du serveur et une unité d'éclairage

- L'unité d'éclairage s'allume automatiquement lorsque quelqu'un entre dans la chambre
- L'unité d'éclairage a un capteur infra-rouge qui détecte le mouvement
- La lumière s'éteint lorsque s'il n'y a pas de détection de mouvement pendant 5 secondes

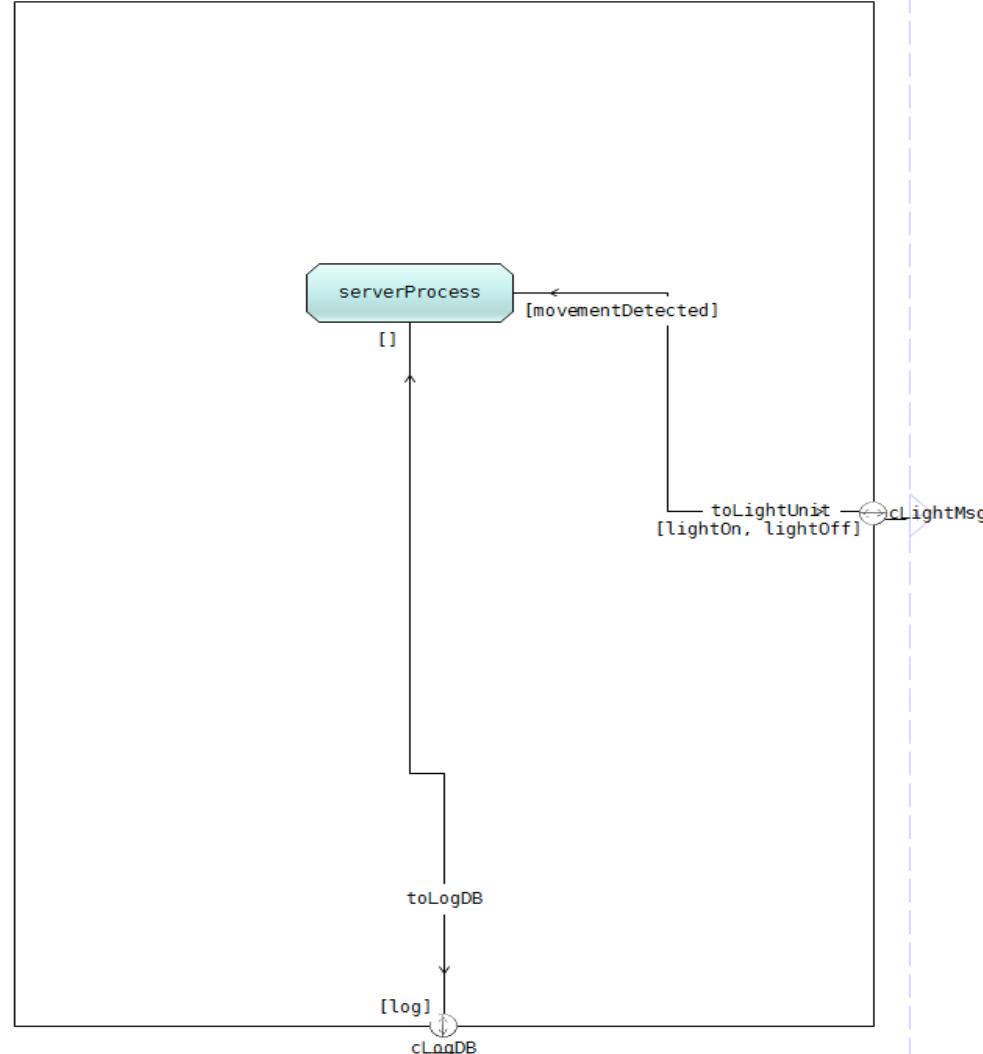
L'INTERACTION ENTRE LE SERVEUR ET L'UNITÉ D'ÉCLAIRAGE



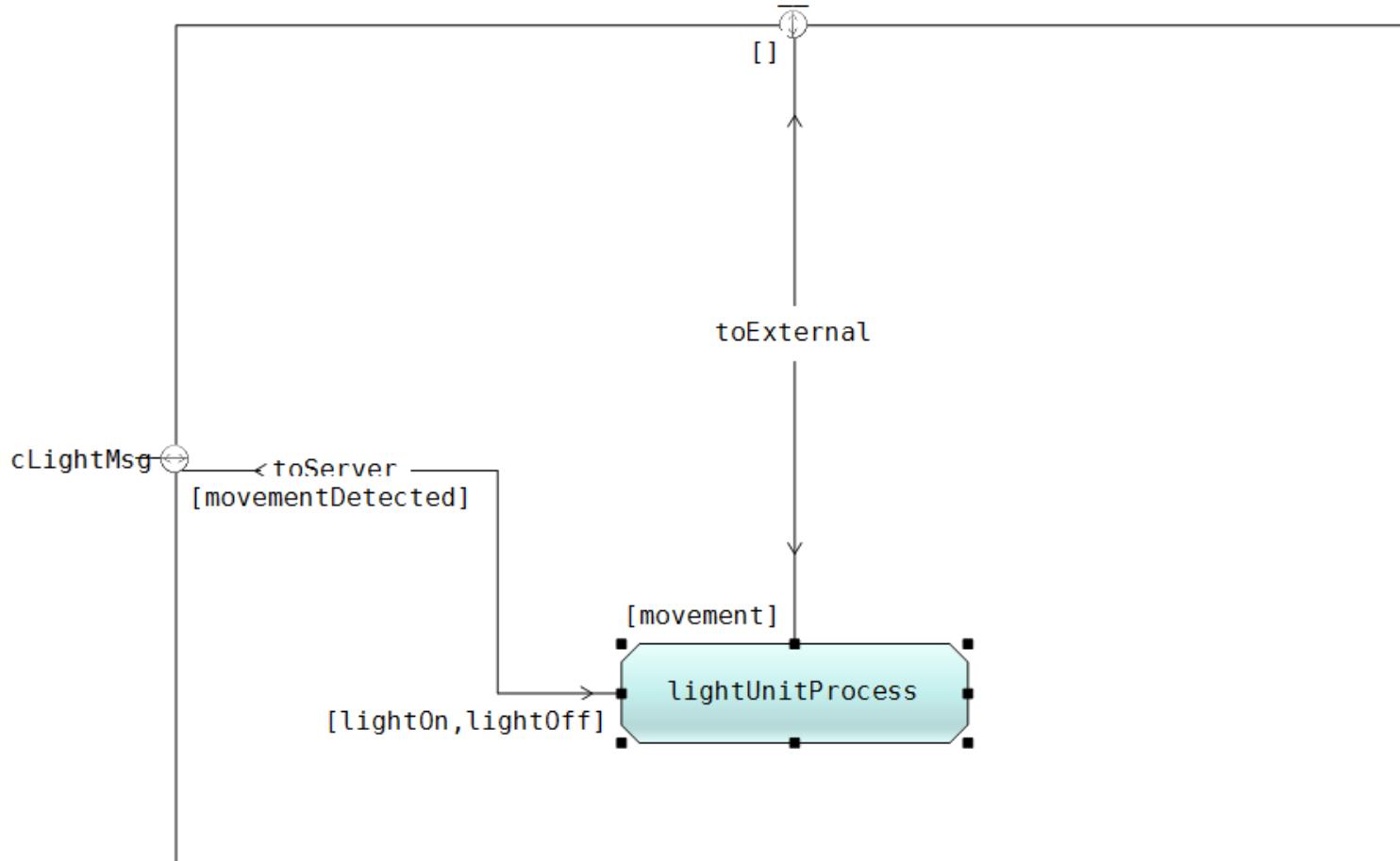
DÉFINITION DU SYSTÈME



DÉFINITION DU BLOC DE SERVEUR

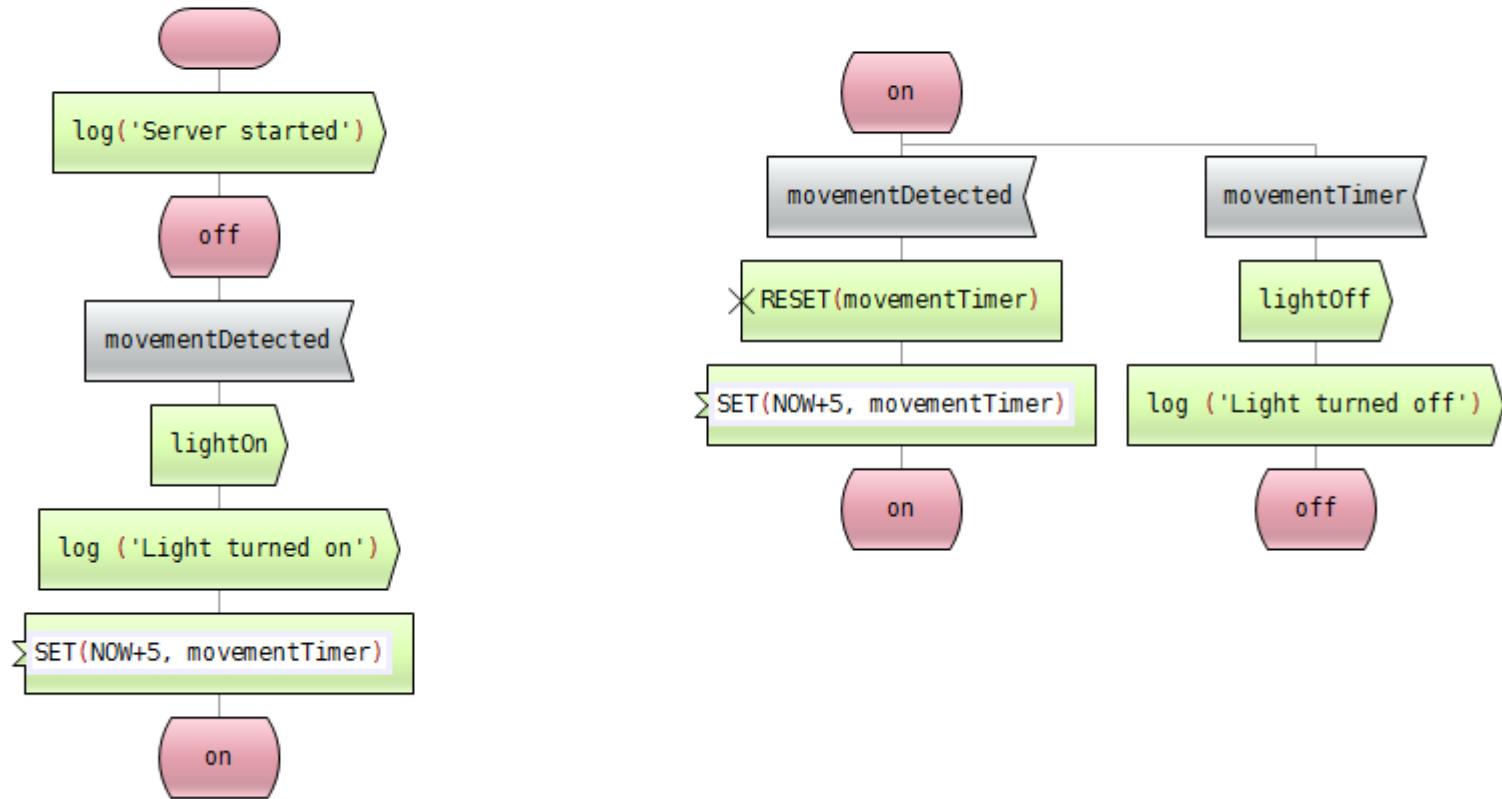


BLOC DE L'UNITÉ D'ÉCLAIRAGE

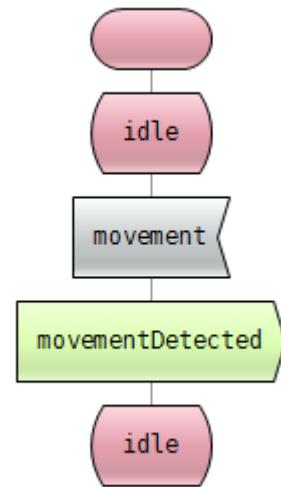


DÉFINITION DU SERVEUR DE PROCESSUS

```
TIMER  
movementTimer;
```



PROCESSUS DE L'UNITÉ D'ÉCLAIRAGE



MERCI!

QUESTIONS?