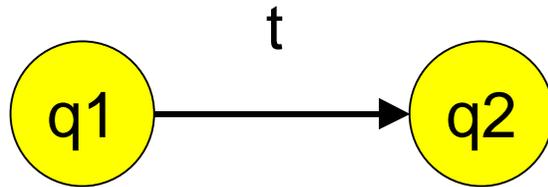


# STATECHARTS

Modèle de comportement

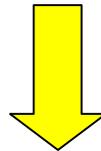
# États/transitions



Machines à états finis  
(FSM)

FSM

- + Hiérarchie
- + Parallélisme
- + Formes de synchronisation



Statecharts (David Harel)

# Comportement

Comportement :

fait référence à la façon dont les choses **changent**.

**Aspect fonctionnel** du comportement :

ce que fait l'objet (QUOI)

**Contraintes** :

- **Fonctionnelles** : Limitations d'utilisation  
Invariants de pre/post conditions
- **Non fonctionnelles** : QoS (Quality-of-Service)  
Qualifie la façon de remplir  
les fonctionnalités

# Comportement (2)

Contraintes fonctionnelles :

souvent exprimées par des FSM

Contraintes non fonctionnelles :

- performances (dont les temporelles)
- précision

Expression : OCL, logiques temporelles, ...

Pour les systèmes temps réel à la fois des contraintes fonctionnelles et non fonctionnelles.

# Comportement (3)

Types de comportement rencontrés en contrôle :

**Comportements simples** : sans mémoire

- fonctions mathématiques
- systèmes combinatoires
- retour de valeur (capteurs, compteurs, ...)

**Comportements continus** : les réactions dépendent du présent ET du passé (notion d'état). Toutefois il n'est pas facile de distinguer des états discrets.

- équations différentielles
- ensembles flous
- réseaux neurones

# Comportement (4)

## Comportements réactifs :

Notion d'état (mémoire)

États distinguables et disjoints

Un état représente une condition qui persiste dans le système pendant un certain temps.

Les états sont **distinguables** par

- les réactions aux événements acceptés
- les changements d'état résultant
- les actions exécutées

Les états sont **disjoints** : le système se trouve à chaque instant dans un et un seul état.

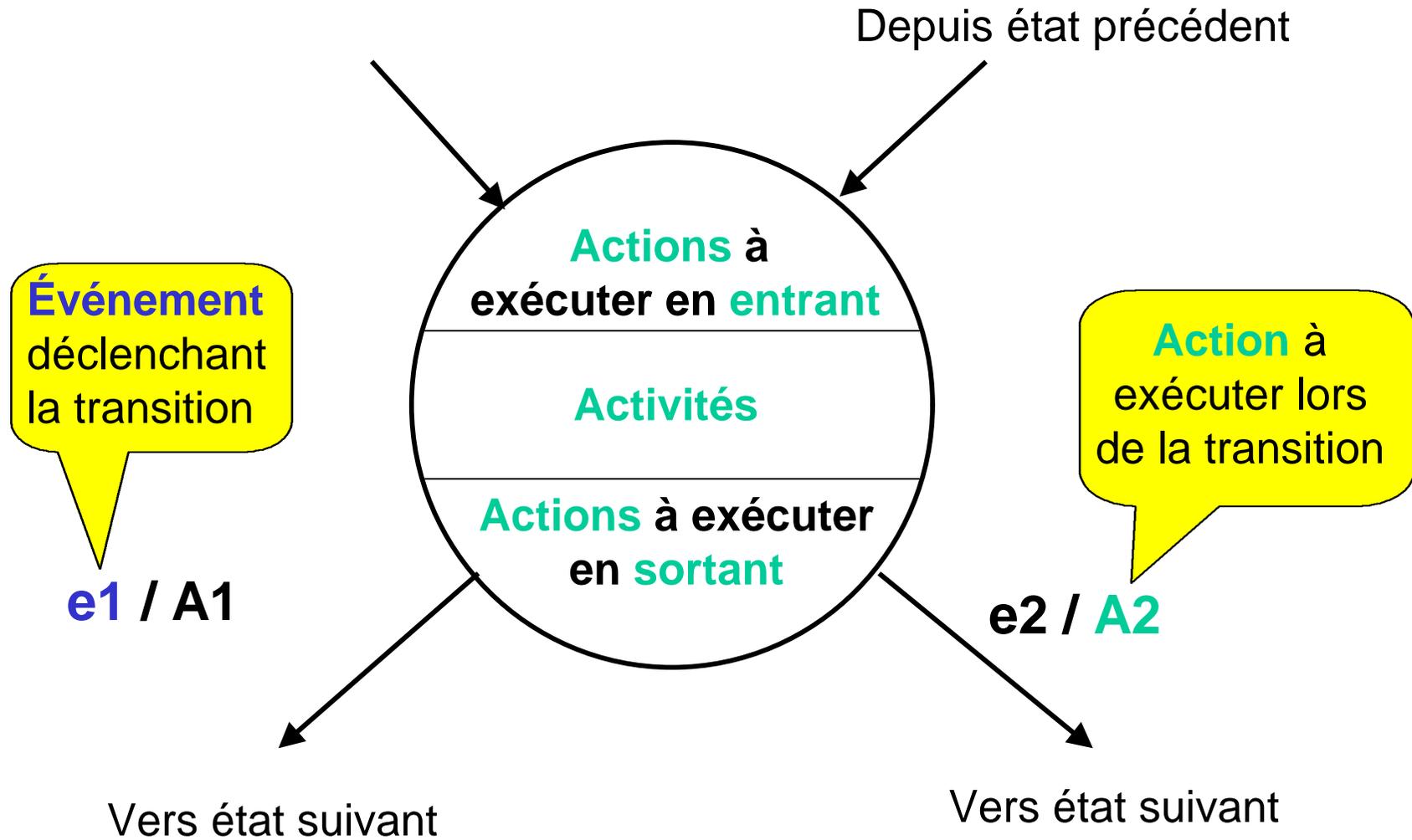
Une **transition** est la réponse à un événement qui produit un changement d'état.

# Machine à états finis

FSM = modèle à états avec simplifications  
⇒ réduction de la complexité

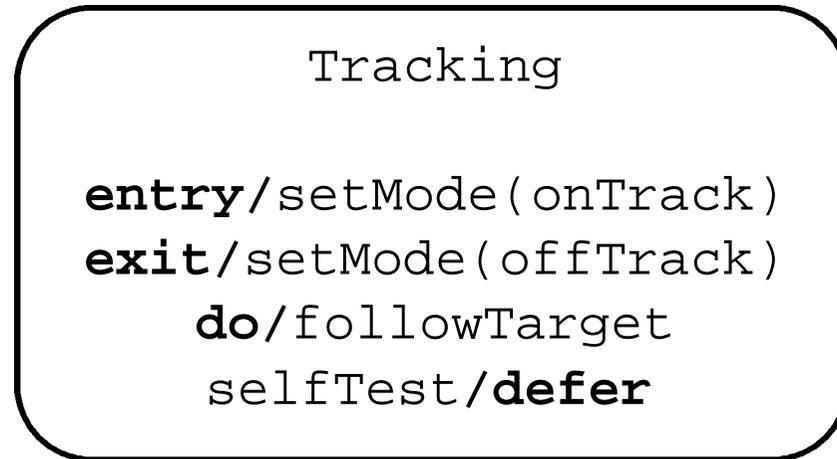
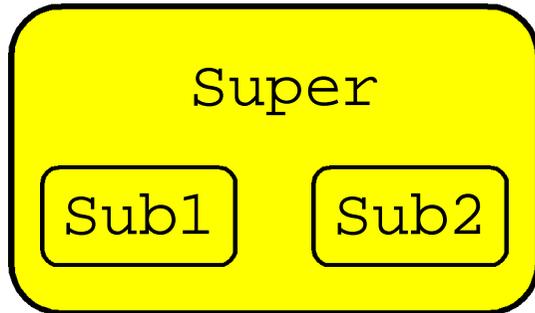
- nombre fini d'états
- comportement invariant dans un état
- le temps est passé dans les états (les transitions sont de durée nulle)
- les transitions sont en nombre fini et bien définies quant à leurs effets

# État





# États



**Nom**

**Action**

**entry**

**exit**

**Activité**

**Sous-états**

**Événements retardés**

chaîne de caractères (facultatif)

exécutée lors de l'entrée

exécutée lors de la sortie

exécutée tant que dans l'état

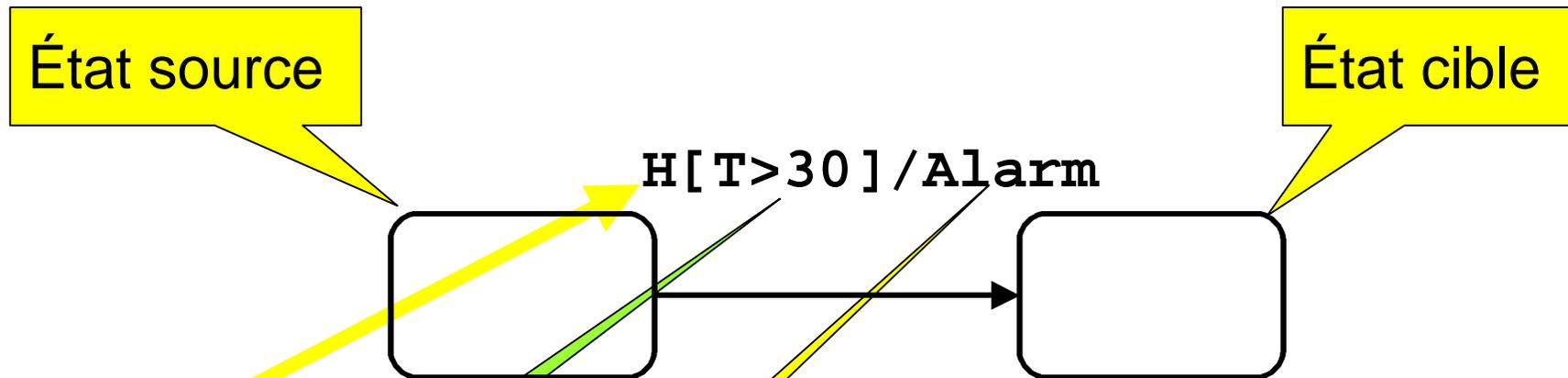
inclus (or-states ou and-states)

jusqu'à la sortie de l'état

# Actions/Activités

- **Action**
  - Est un comportement élémentaire
  - Durée d'exécution : négligeable
  - Exécution atomique, non interruptible
  - Actions d'entrée : **entry/**
  - Actions de sortie : **exit/**
- **Activité**
  - Exécution qui dure tant que l'état est actif
  - Interruptible
  - Syntaxe : **do/**

# Transition



**Événement déclencheur**

(peut être absent)

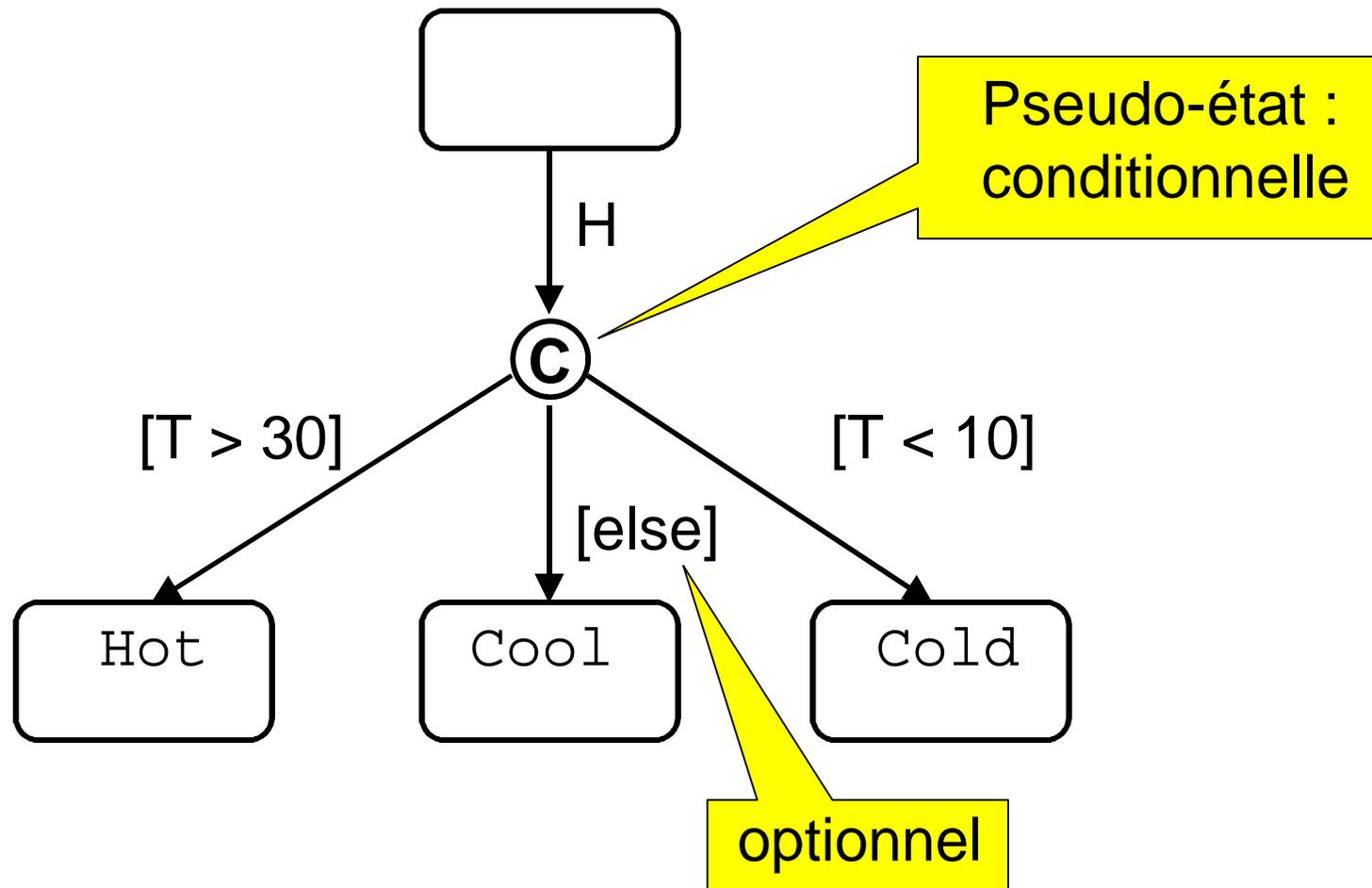
**Garde**

condition nécessaire pour transiter lorsque le déclencheur est présent (facultatif)

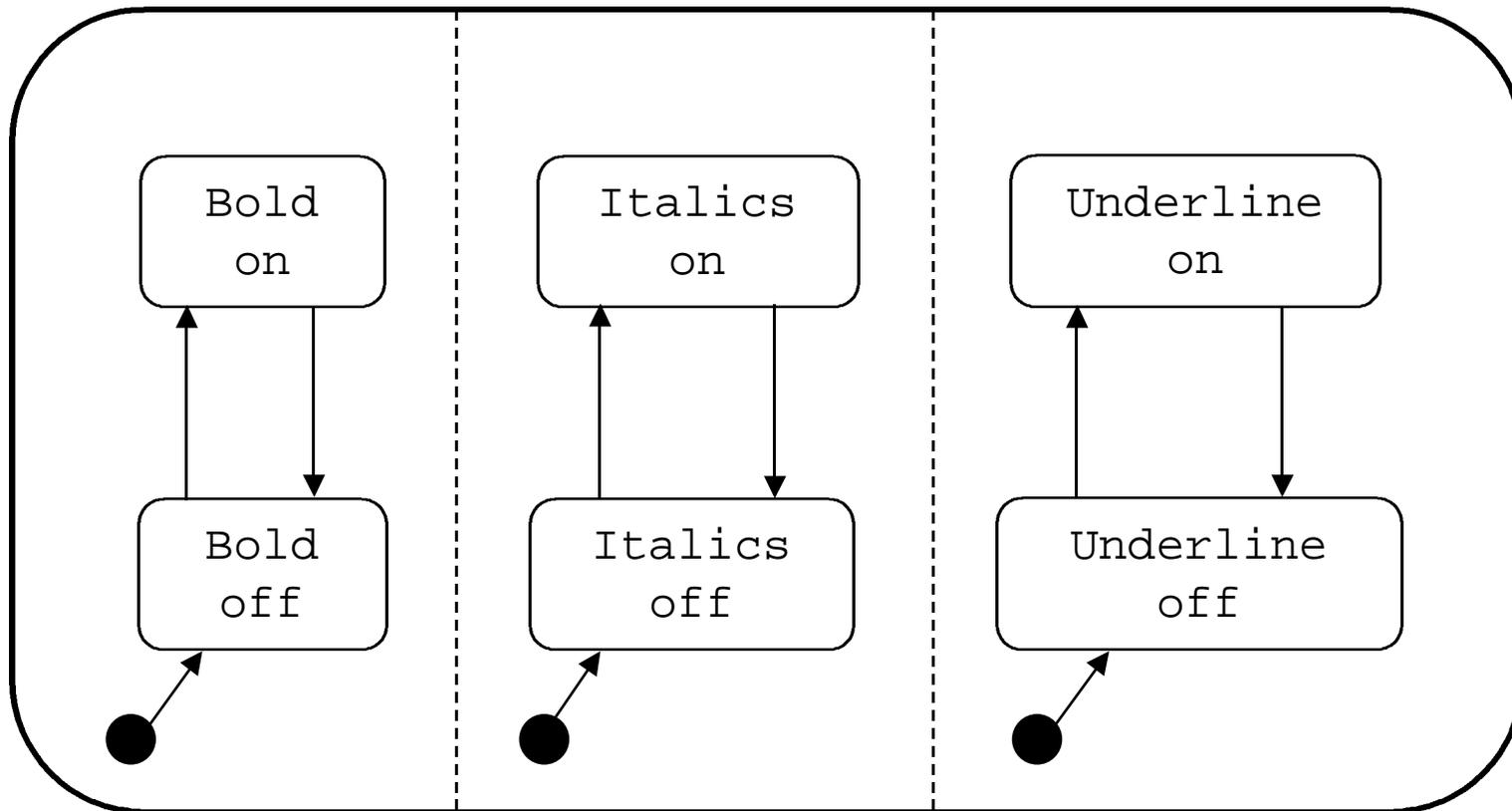
**Action**

exécution **atomique**, peut être une séquence d'actions

# Transition (2)



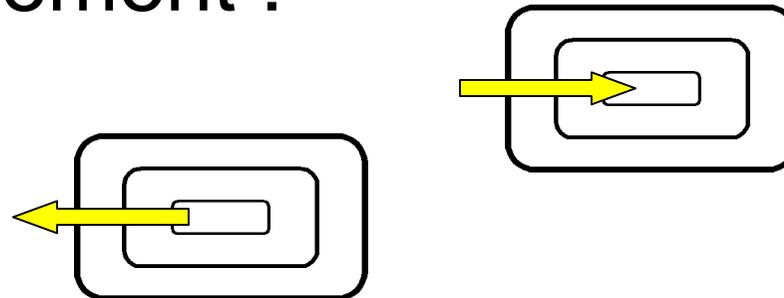
# États concurrents



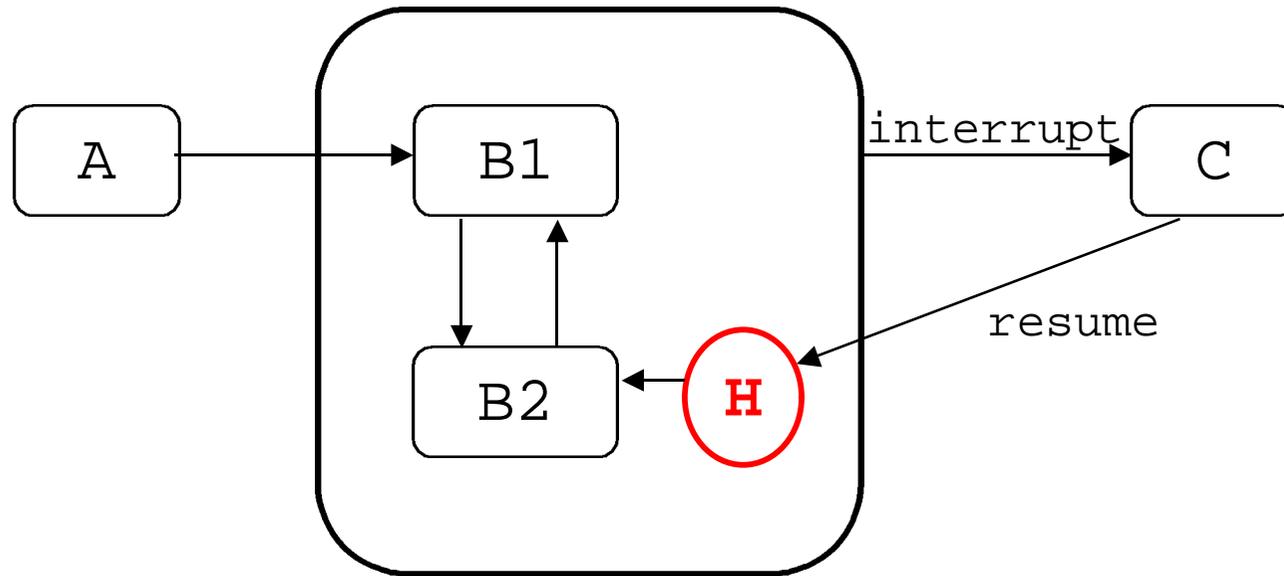
$2 \times 3$  au lieu de  $2^3$

# Exécution

- Notion d'**instant** : rtc (run-to-completion)  
model steps
- Transition : exécution **atomique** et de **durée négligeable**
  1. *Exit actions* de l'état source
  2. *Transition actions*
  3. *Entry actions* de l'état cible
- En cas de raffinement :
  - En entrant :
  - En sortant :



# Entrée par histoire



$H^*$  indique un appel récursif

# Exemple\* : micro-ondes

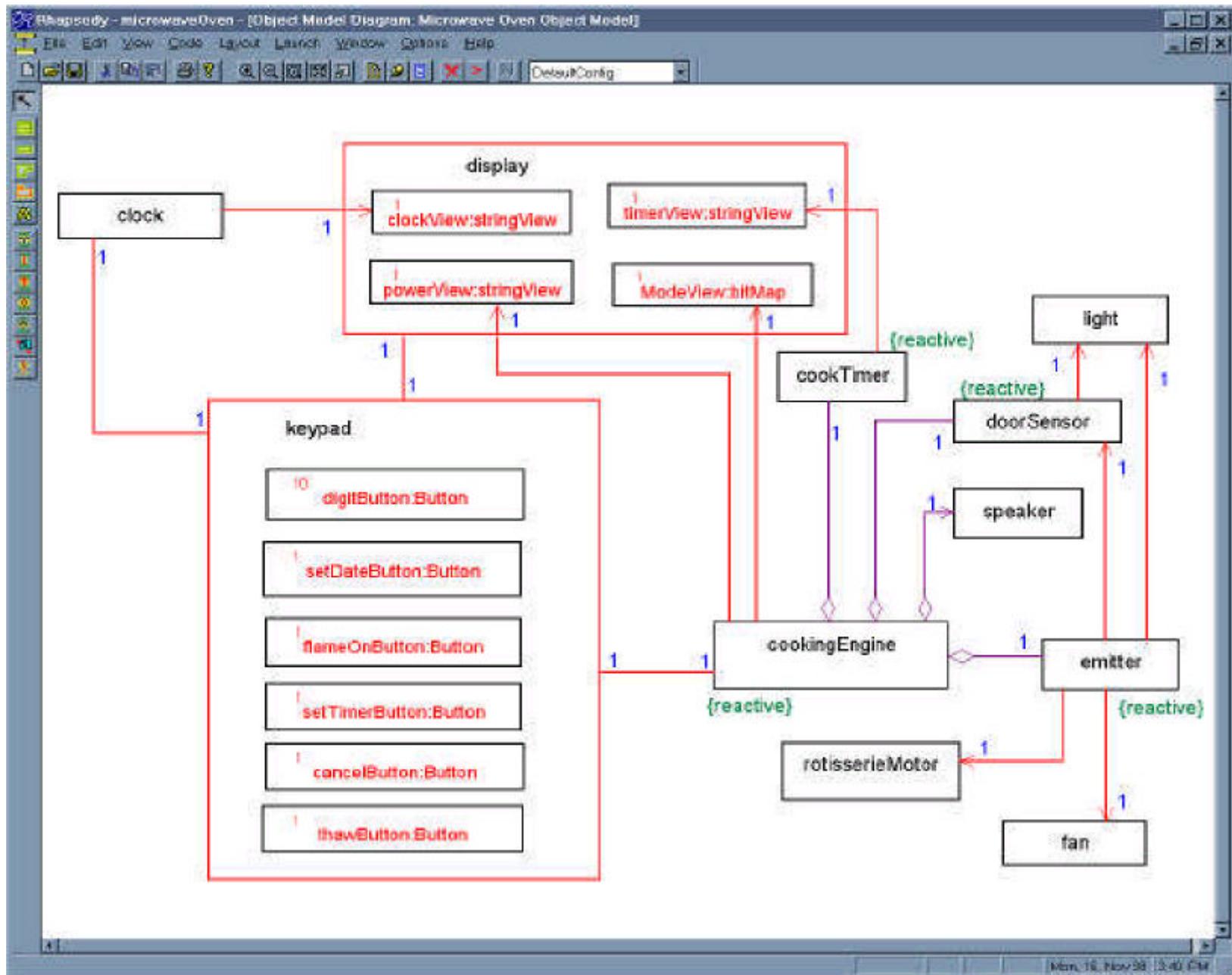
- Mode cuisson : puissance et durée
- Puissance réglée par PWM
- L'émetteur n'est mis en route que si la porte est fermée
- La lumière s'allume quand
  - la porte est ouverte
  - la cuisson démarre
  - en mode décongélation
- Mode minuterie
- .... *\*Adapté d'un article B.P. Douglass (I-Logix)*

# Micro-ondes (2)

## Diagramme de classes

4 classes réactives :

- cookingEngine
- cookTimer
- emitter
- doorSensor

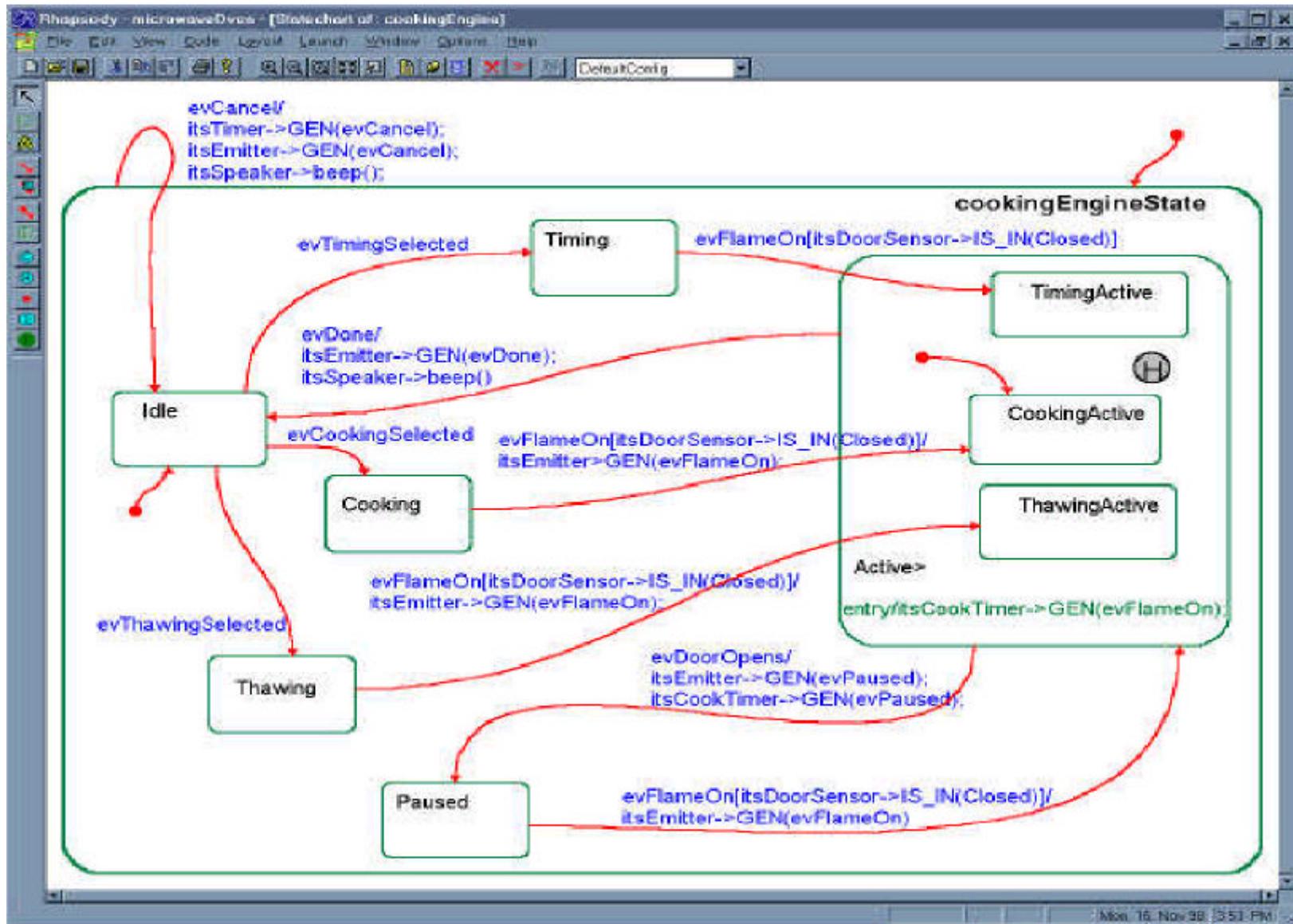


# Micro-ondes (3)

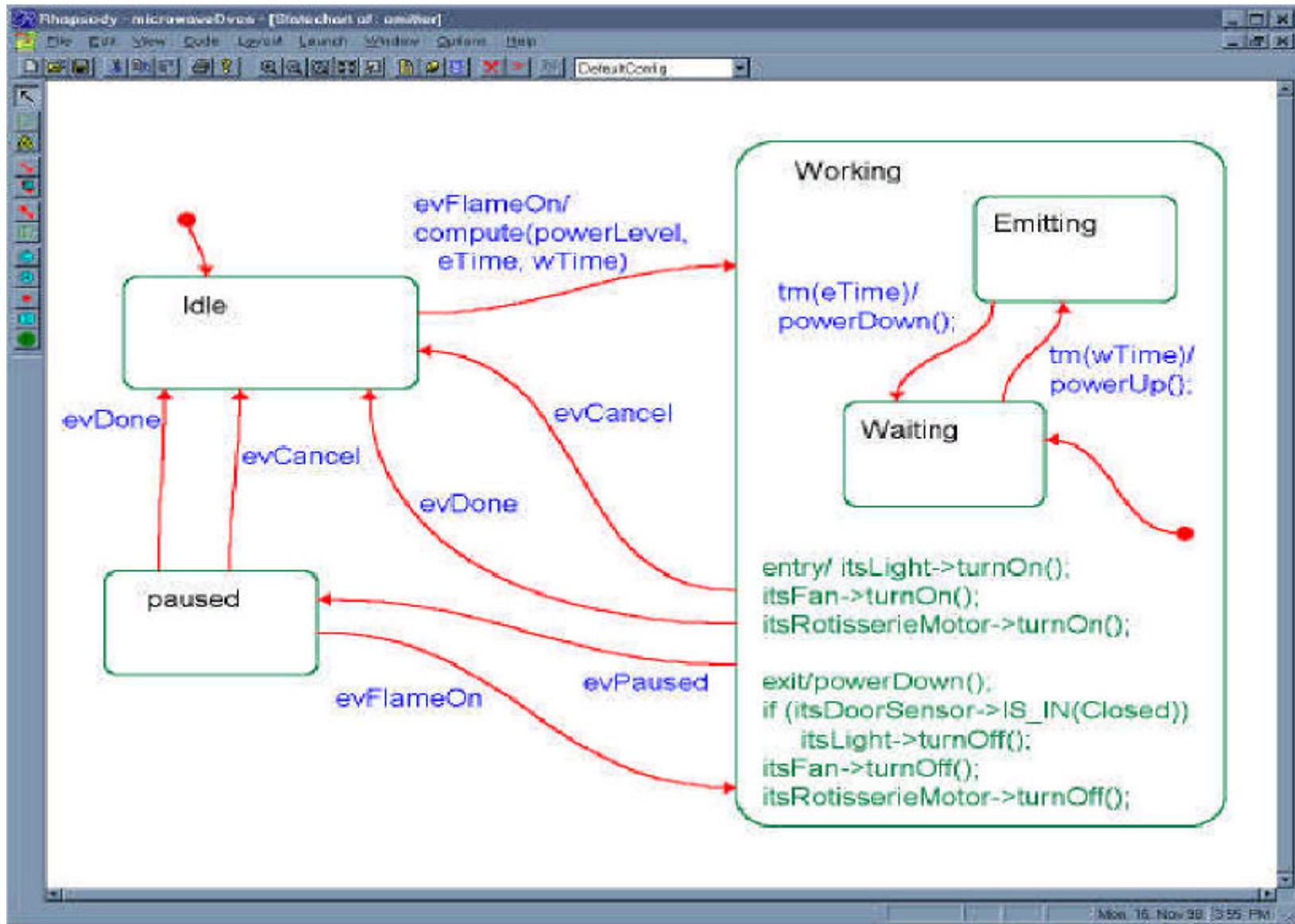
## Comportement des classes réactives :

- Statechart du `cookingEngine`
- Statechart de l'`emitter`
- Statechart du `cookTimer`
- Statechart du `doorSensor`

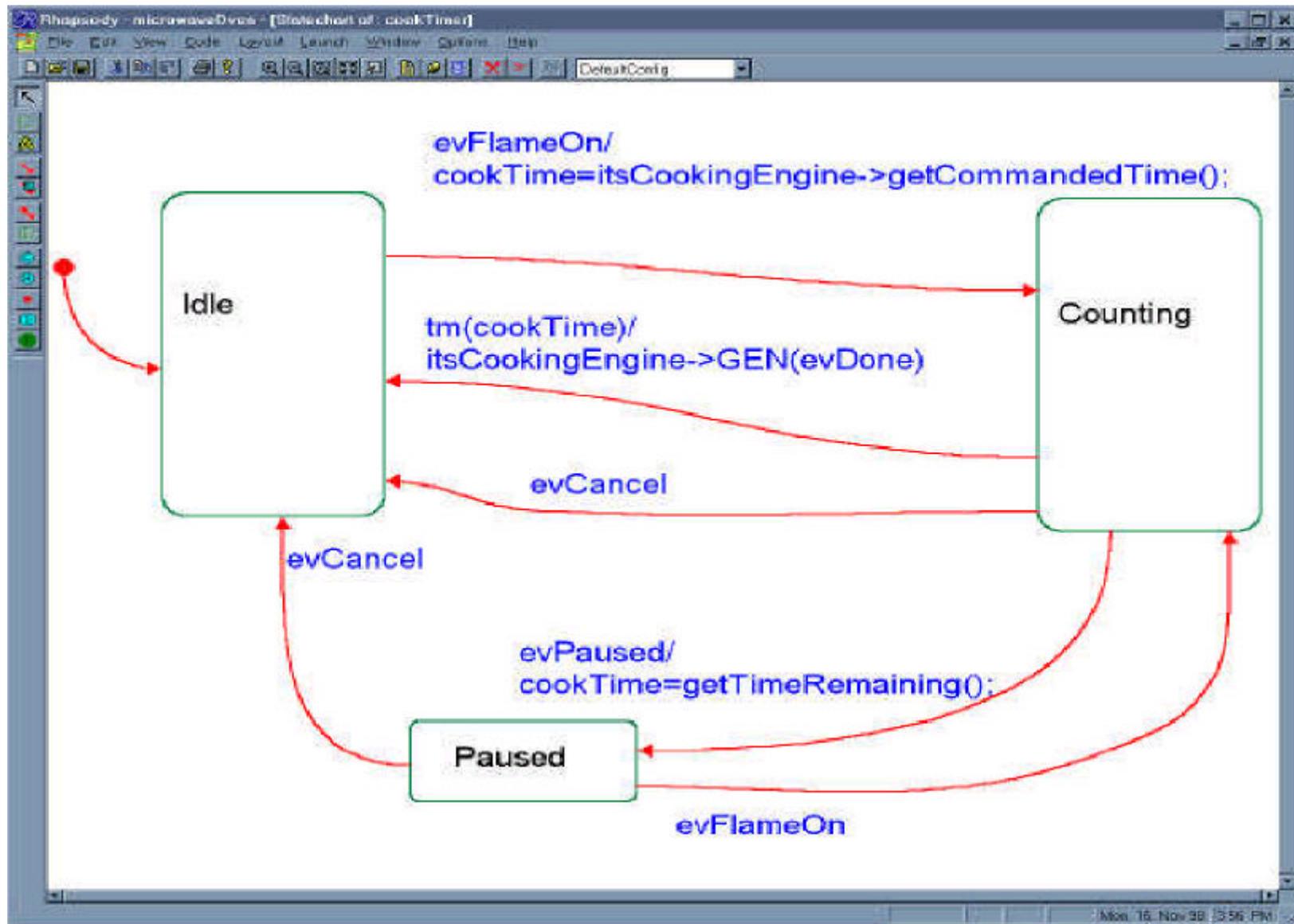
# cookingEngine



# emitter



# cookTimer



# doorSensor

