

D

Simulateurs abstraits pour DEVS

D.1 Simulateur abstrait pour un modèle DEVS atomique

Nous présentons le ici le simulateur abstrait d'un modèle atomique.

Le signe « := » est l'affectation, « Fonct_Sortie » correspond à λ , « Trans_int » à δ_{int} et « Trans_ext » à δ_{ext} dans un modèle DEVS atomique. « // » précède un commentaire (d'après [ZKP00]).

```
1  DEBUT simulateur devs

      DEBUT déclaration variables
      parent // le coordinateur
5     S      // l'ensemble des états
      t1     // la date d'entrée dans l'état s
      tn     // la date du prochain évènement
      devs   // le nom du modèle DEVS associé au simulateur
      e      // le temps passé dans l'état s
10    (s,e)  // (s,e) l'état total de ce modèle
      y      // la valeur courante de sortie du modèle
      FIN déclaration variables

      QUAND réception d'un évènement d'initialisation (i,t)
15    t1 := t
      tn := t1 + ta(s)
      FIN QUAND

      QUAND réception d'un évènement de transition externe : (x,t)
20    SI t1 <= t <= tn ALORS
      e := t - t1
      s := Trans_ext(s,e,x) // calcul de la fonction de transition externe
      // x désigne le couple (port,valeur)

      t1 := t
25    tn := t1+ta(s)
      SINON erreur
      FIN SI
      FIN QUAND
```

```

30  QUAND réception d'un évènement de transition interne (*,t)
      SI t = tn ALORS
          y := Fonct_Sortie(s)
          Envoyer au parent l'évènement de sortie (y,t)
          s := Trans_int(s)
35      t1 := t
          tn := t1+ta(s)
      FIN SI
      SINON erreur
40  FIN QUAND
FIN Simulateur devs

```

D.2 Simulateur abstrait pour un modèle DEVS couplé

Nous présentons le simulateur abstrait d'un coordinateur DEVS. Ce simulateur correspond au modèle DEVS couplé. Nous avons adapté l'algorithme décrit dans [ZKP00] puisque nous n'utilisons pas la fonction de sélection. De plus, pour une question de lisibilité, nous avons utilisé les notations formelles au minimum.

« := » est le signe d'affectation, « // » précède un commentaire (algorithme adapté d'après [ZKP00]).

```

1  DEBUT coordinateur

      DEBUT déclaration variables
          parent // le coordinateur parent
          D // ensemble des modèles composants
5      IC // ensemble des connexions internes
          EOC // ensemble des connexions externes en sortie
          EIC // ensemble des connexions externes en entrée
          t1 // la date du dernier évènement
10     tn // la date du prochain évènement
          Echancier // liste triée des couple (d,tn) d appartient à D
      FIN déclaration variables

      QUAND réception l'évènement d'initialisation (i,t)
15     Envoyer à D l'évènement d'initialisation (i,t)
          Trier Echancier // en fonction de tn de d appartient à D
          t1 := maximum(t1 de d) // d appartient à D
          tn := minimum(tn de d) // d appartient à D
      FIN QUAND
20
      QUAND réception d'un évènement de transition interne (*,t)

```

```

    SI t = tn ALORS
      d = dépiler Echéancier
      Envoyer l'évènement (*,t) à d
25      Trier Echéancier          // en fonction de tn de d appartient à D
      t1 := maximum(t1 de d) // d appartient à D
      tn := minimum(tn de d) // d appartient à D
    SINON erreur
    FIN SI
30  FIN QUAND

  QUAND réception d'un évènement de transition externe (x,t)
      // x désigne le couple (port,valeur) en entrée
    SI t1 <= t <= tn ALORS
35      POUR d dans EIC associé à x
          Envoyer l'évènement (x,t) à d
      FIN POUR
      Trier Echéancier          // en fonction de tn de d appartient à D
      t1 := maximum(t1 de d) // d appartient à D
40      tn := minimum(tn de d) // d appartient à D
    SINON erreur
    FIN SI
  FIN QUAND

45  QUAND réception d'un évènement de sortie (y,t) du modèle d
      // d appartient à D
      // y désigne le couple (port,valeur)
      // en sortie d'un modèle composant
    POUR i dans EOC associé à y // i appartient à D
50      Envoyer l'évènement de sortie (y,t) à parent
    FIN POUR

    POUR i dans IC associé à y // i appartient à D
      Envoyer l'évènement de transition externe (x,t) au modèle i
55  FIN POUR
  FIN QUAND

  FIN coordinateur

```

D.3 Simulateur abstrait pour le coordinateur racine

Le coordinateur racine correspond à la boucle générale de simulation d'un modèle DEVS.

« := » est le signe d'affectation, « // » précède un commentaire (d'après [ZKP00]).

```
1  DEBUT coordinateur racine

    DEBUT déclaration variables
        t      \\ date courante de la simulation
5      enfant \\ simulateur ou coordinateur subordonné direct
    FIN déclaration de variables

    t := t0    \\ intialisation du temps

10   Envoyer évènement d'initialisation (i,t) à Enfant
    t := tn de Enfant \\ réception de (Prochain,tn)

    TANT QUE t <= durée de la simulation FAIRE
        Envoyer l'évènement de transition interne (*,t) à enfant
15      t := tn de Enfant
    FIN TANT QUE
FIN coordinateur racine
```