

1 DEVOIR 1 : CONSTRUIRE UN MODELE D'UN REVEILLE-MATIN

Laboratoire présenté à Dr. G. Bochmann
dans le cadre du cours SEG2506 : Construction de logiciel

Par Ryan Germain (3335821) - rgerm026@uottawa.ca
et Pascal Tozzi (3567599) – ptozz104@uottawa.ca

Université d'Ottawa, Département de génie
le 5 février 2006

Soumission :

2/2% Page-titre : académique complète

3/3% Projet SDL (assez pour faire la simulation / validation – tous les MSC's inclus)

Notes suivantes à réviser et confirmer:

Rapport et projet :

1. [=22.5/25%] préparation :
 1. 2.5/5% Dictionnaire des termes
 2. 5/5% Modèle conceptuel UML du système et de son environnement
 3. 5/5% Description des exigences que vous adoptez pour le système et ses composantes, ainsi que les hypothèses. En particulier, vous devriez préciser l'interface du réveille-matin par lequel l'utilisateur peut contrôler le fonctionnement de l'appareil.
 4. 10/10% Quelques cas d'utilisation. Pour chaque cas d'utilisation, choisissez un nom, fournissez une description informelle de quelques phrases (en français) et un diagramme MSC (qui peut être écrit à la main).
2. [=35/35%] Construire un modèle fonctionnel du réveille-matin (conforme aux exigences ci-dessus) en SDL
 1. 5/35 : donné, ne fonctionne pas, difficile à suivre
 2. 5-10 / 35 : donné, ne fonctionne pas ET évidemment incomplet
 3. 15/35 : ne fonctionne mais facile à lire et semble complet
 4. 25/35 : fonctionne mais par selon le test
 5. 35/35 : fonctionne selon le test
3. [=15/15%] Description des simulations exécutées: des activités de vérification que vous avez faites, quelles erreurs ont été initialement détectées et ensuite corrigées; indiquez si la vérification a finalement été un succès, etc.....
 1. 5% S.v.p. donner une courte description (une demie à une page de texte) des activités de vérification
 2. 7% quelles erreurs ont été initialement détectées et ensuite corrigées
 3. 3% indiquez si la vérification a finalement été un succès, etc.
4. [=20/20%] Fournir (sur disquette) au moins un MSC généré par l'outil TAU pendant les simulations; il devrait correspondre à un cas d'utilisation mentionné ci-haut .
 1. 5/20 : correspond, mais ne fonctionne pas OU trop simple
 2. 20/20 : correspond, fonctionne ET de taille raisonnable
 3. 10/20 : ne correspond pas mais fonctionne ET de taille raisonnable
1. 0/20 : ne correspond pas ET ne fonctionne pas

Bonus pour effort spécial sur le rapport : 2.5 points

Jusqu'à présent : $\frac{2}{2} + \frac{22.5}{25} + \frac{15}{15} = \frac{39.5}{42}$; $\circ 2.5 = \frac{42}{42} + 58/58 \Rightarrow 100\%$

DEDUCTIONS :

- * 5% qualité de la langue
- * 5% pas de description sur les figures etc.
- 10% cas d'utilisation et MSC pas complets selon la description ET usage habituel

Introduction :

Ce devoir consiste à construire un modèle SDL d'un réveille-matin. Pour ce faire, nous avons défini des exigences précises à partir de ceux qui ont été fournies, construit un modèle UML du réveille-matin et de son environnement, développer un modèle en SDL puis finalement le modèle a été simulé pour vérifier son fonctionnement.

Dictionnaire des termes :

Alarme : un bruit ou signal qui permet d'avertir l'utilisateur à un  donné
Alarme déclenchée : une alarme qui est présentement dans un  l'utilisateur
Temps courant : temps affiché par le réveille-matin
Snooze : fonction qui permet de retarder l'alarme de 5 minutes lorsque ce dernier est déclenché
Intensité de la sonnerie : Niveau sonore de la sonnerie
Temps de l'alarme : temps à lequel l'alarme se déclenche

Descriptions des exigences :

Voici les exigences initiales qui ont été fournies :

- Le réveille-matin affiche normalement le temps courant (en heures et minutes). Initialement, cette valeur est 00:00.
- Le temps courant peut être ajusté par l'utilisateur. Pour simplifier ce processus, l'interface devrait permettre le changement des heures et des minutes, indépendamment, et il devrait être possible d'avancer les minutes par 10 unités à la fois.
- Pendant que le temps de l'alarme est ajusté par l'utilisateur, le temps de l'alarme est affiché, au lieu du temps courant.
- Le réveille-matin contient un radio et une sonnerie; l'utilisateur peut choisir lequel devrait fournir le bruit pour le réveil. La sonnerie a trois niveaux d'intensité.
- Quand l'alarme est déclenchée, l'utilisateur peut arrêter complètement, ou peut actionner la touche « snooze » qui a pour effet que l'alarme s'arrête immédiatement, mais reprendra de nouveau dans 5 minutes.
- Si l'alarme n'est pas arrêtée par l'utilisateur, elle reste active pendant deux minutes. Si le bruit de la sonnerie a été sélectionné par l'utilisateur, alors l'intensité de la sonnerie pendant la première minute sera comme choisie par l'utilisateur, et pendant la deuxième minute, elle sera à un niveau sonore plus élevé (si cela est possible).

Voici la description des exigences que nous avons adoptées pour le système :

1. L'interface du réveille-matin doit contenir :
 - 1.1. un bouton « alarm »
 - 1.2. un bouton « time »
 - 1.3. un bouton « minute »
 - 1.4. un bouton « minute + 10 »
 - 1.5. un bouton « hour »
 - 1.6. un bouton « snooze »

- 1.7. un bouton « sound »
- 1.8. un bouton « volume »
2. Le réveille-matin doit afficher le temps courant (en heures et minutes), sauf si l'utilisateur change le temps courant ou le temps de l'alarme.
3. Initialement, le temps courant doit être 00:00.
4. Le temps affiché doit toujours être en mode 24h.
5. Le temps courant doit avoir la possibilité d'être ajusté par l'utilisateur.
6. L'interface doit permettre le changement des heures et des minutes, indépendamment.
7. Pour ajuster le temps courant, l'utilisateur doit d'abord appuyer sur le bouton « time », le garder appuyé, ensuite appuyer sur les boutons « minute », « minute + 10 » ou « hour » pour ajuster le temps et finalement lâcher le bouton « time ».
8. Pendant que l'utilisateur ajuste le temps courant, s'il appuie le bouton « minute », le temps courant doit augmenter d'une minute.
9. Pendant que l'utilisateur ajuste le temps courant, s'il appuie le bouton « minute + 10 », le temps courant doit augmenter de dix minutes.
10. Pendant que l'utilisateur ajuste le temps courant, s'il appuie le bouton « hour », le temps courant doit augmenter d'une heure.
11. Le temps de l'alarme doit avoir la possibilité d'être ajusté par l'utilisateur.
12. Le temps de l'alarme doit être affiché, au lieu du temps courant, pendant que le temps de l'alarme est ajusté.
13. Pour ajuster le temps de l'alarme, l'utilisateur doit d'abord appuyer sur le bouton « alarm », le garder appuyé, ensuite appuyer sur les boutons « minute », « minute + 10 » ou le bouton « hour » pour ajuster le temps et finalement lâcher le bouton « alarm ».
14. Pendant que l'utilisateur ajuste le temps de l'alarme, les mêmes boutons pour ajuster le temps auront les effets précisés comme dans les exigences 8 à 10, sauf que ce sera le temps de l'alarme qui sera modifié.
15. Le réveille-matin doit contenir une radio.
16. Le réveille-matin doit contenir une sonnerie.
17. L'utilisateur doit avoir la possibilité de choisir le bruit pour l'alarme en appuyant le bouton « sound », qui échange entre les états de radio, sonnerie et « off ».
18. Initialement, l'état de « off » doit être sélectionné.
19. L'alarme ne doit pas sonner si le commutateur « sound » est à « off ».
20. La sonnerie doit avoir trois niveaux d'intensité :
 - 20.1. Basse intensité
 - 20.2. Intensité moyenne
 - 20.3. Haute intensité
21. L'utilisateur doit avoir la possibilité de changer l'intensité de la sonnerie.
22. Initialement, la sonnerie doit être à haute intensité.
23. Lorsque l'utilisateur appuie le bouton « volume » et l'intensité de la sonnerie est à « haute intensité », l'intensité de la sonnerie changera à « basse intensité ».
24. Lorsque l'utilisateur appuie le bouton « volume » et l'intensité de la sonnerie est à « basse intensité », l'intensité de la sonnerie changera à « intensité moyenne ».
25. Lorsque l'utilisateur appuie le bouton « volume » et l'intensité de la sonnerie est à « intensité moyenne », l'intensité de la sonnerie changera à « haute intensité ».
26. L'utilisateur doit avoir la possibilité d'arrêter l'alarme, en appuyant le bouton « alarme », lorsque l'alarme est déclenchée.

27. L'utilisateur doit avoir la possibilité d'appuyer le bouton « snooze » qui a pour effet que pendant que l'alarme est déclenchée, l'alarme s'arrête immédiatement, mais reprendra de nouveau dans 5 minutes.
28. L'alarme déclenchée doit rester active pendant deux minutes, à moins d'être interrompu par l'utilisateur qui appuie le bouton « snooze » ou « alarme ».
29. L'intensité de la sonnerie pendant la première minute doit être comme choisie par l'utilisateur.
30. Pendant la deuxième minute de l'alarme, l'intensité sera changée à un niveau sonore plus élevé, à moins que l'intensité soit déjà à l'haute intensité.

Voici les précisions que nous avons ajoutées :

Exigence	Précision
1	Précision de l'interface
2	Référence à la première exigence initiale, précision de « affiche normalement »
4	Précise la méthode de présenter le temps courant, qui affecte comment cette variable est sauvegarder par le système
7-10	Précision de l'interface et des étapes à suivre pour changer le temps courant
14	Précision de l'interface et des étapes à suivre pour changer le temps de l'alarme
23-25	Précision de l'interface pour le volume de la sonnerie
26-29	Précision de l'interface pour arrêter l'alarme

Tableau 1: Précision des exigences

Modèle conceptuel :

Voici un diagramme de classe du réveille-matin que nous avons implémenté :

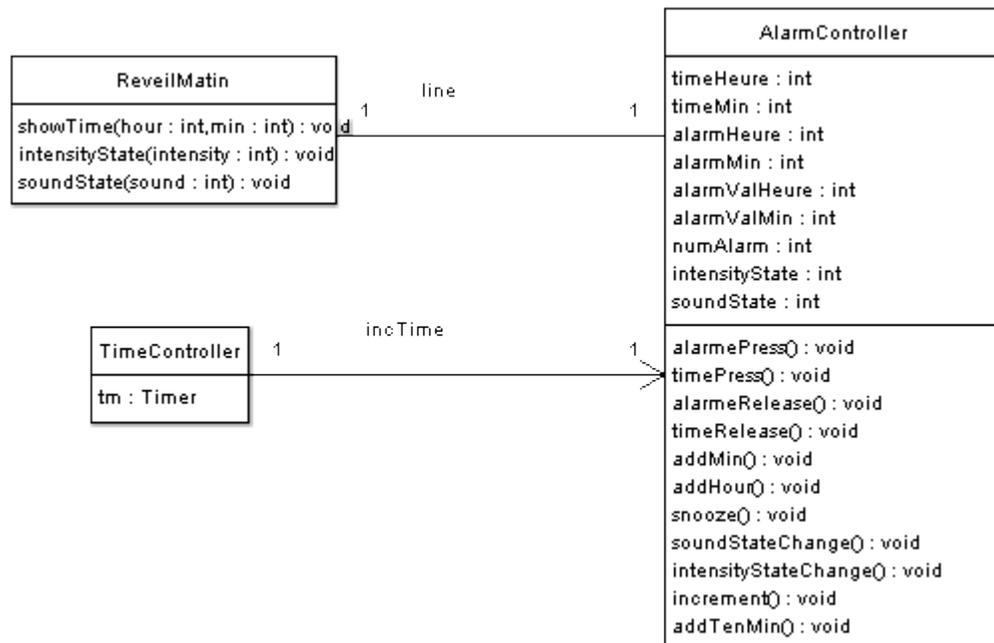


Figure 1: Modèle conceptuel

On peut voir que toutes les variables déclarées avec DCL sont des variables d'instances, les canaux en associations, les processus et les blocs se traduisent en classe et que les signaux se traduisent en méthodes.

Voici un diagramme de cas d'utilisation représentant quelques cas types :

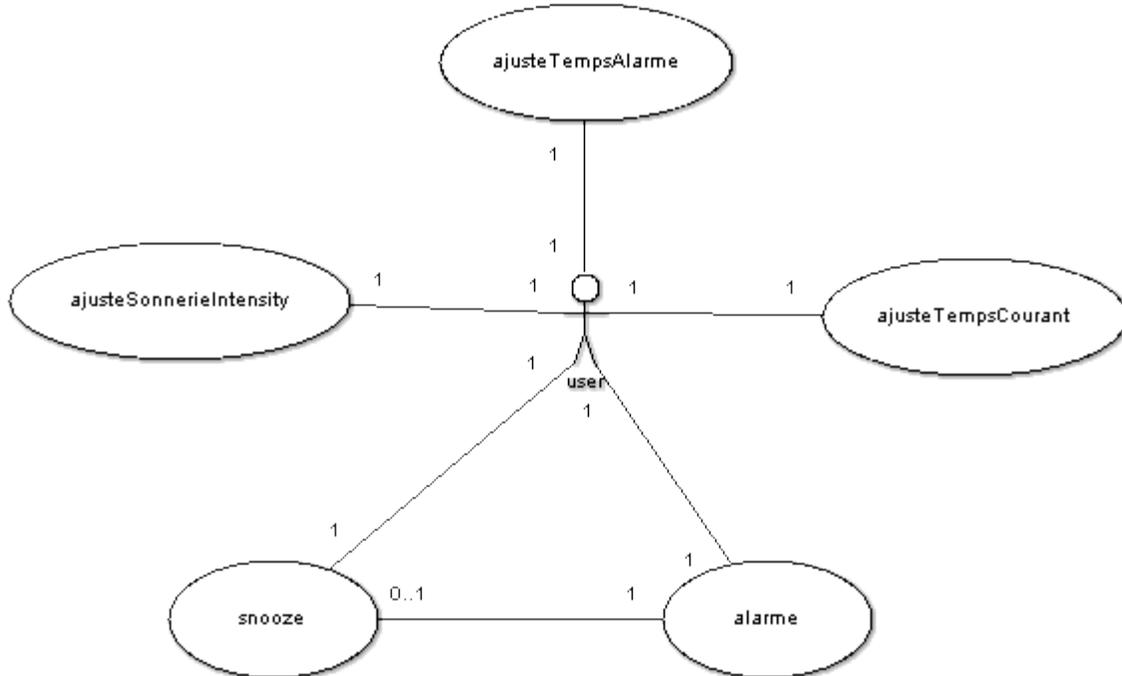


Figure 2: Cas d'utilisation du réveil-matin

Le cas d'utilisation « ajusteTempsCourant » décrit le cas types où l'utilisateur ajuste le temps courant du réveil-matin. Au début, le réveil-matin s'initialise en assignant 00:00 au temps courant (et l'affiche), « off » au bruit de l'alarme et haute intensité pour la sonnerie. De suite, l'utilisateur appuie le bouton « time » et ajuste le temps courant en appuyant les boutons « hour », « minute + 10 » et « minute ». Le diagramme de séquence correspondant à ce cas d'utilisation est sauvegarder dans le fichier « UseCase1.msc ». On peut voir que les minutes et les heures sont incrémentées indépendamment sans affecter l'autre.

Le cas d'utilisation « ajusteTempsAlarme » décrit le cas types où l'utilisateur ajuste le temps de l'alarme. Au début, le réveil matin s'initialise comme le dernier cas-type. Ensuite, le temps courant est incrémenté quelquefois. Par la suite, l'utilisateur appuie le bouton alarme et le temps de l'alarme s'affiche. Vous pouvez remarquer que le temps de l'alarme n'est pas pareil au temps courant. Le diagramme de séquence correspondant à ce cas d'utilisation est sauvegarder dans le fichier « UseCase2.msc ».

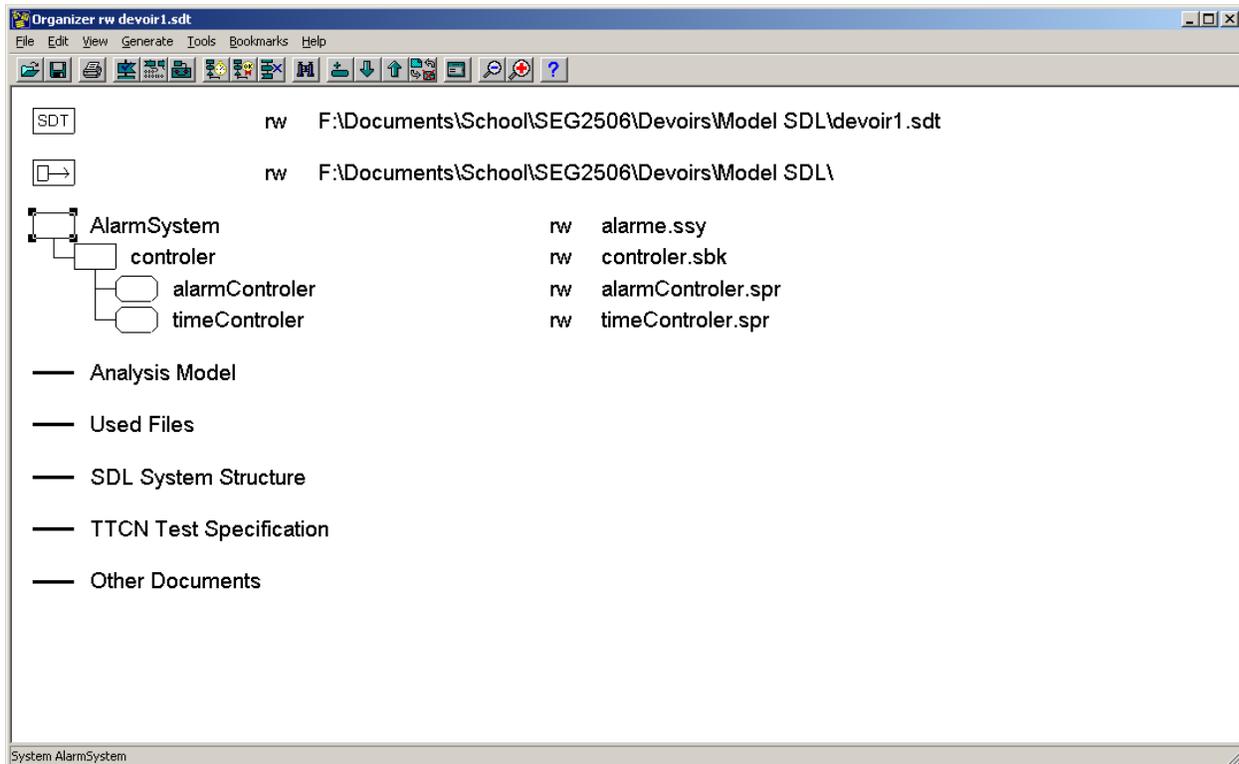
Le cas d'utilisation « ajusteSonnerieIntensity » décrit le cas types où l'utilisateur change le bruit pour l'alarme et l'intensité de la sonnerie. On remarque qu'après l'initialisation, l'état de la sonnerie est à « off ». L'utilisateur peut ensuite inter-changer l'état à « radio », à « sonnerie » et retourner à « off ». L'intensité peut aussi être inter-changer à basse, moyenne et retourner à haute intensité. Le diagramme de séquence correspondant à ce cas d'utilisation est sauvegarder dans le fichier « UseCase3.msc ».

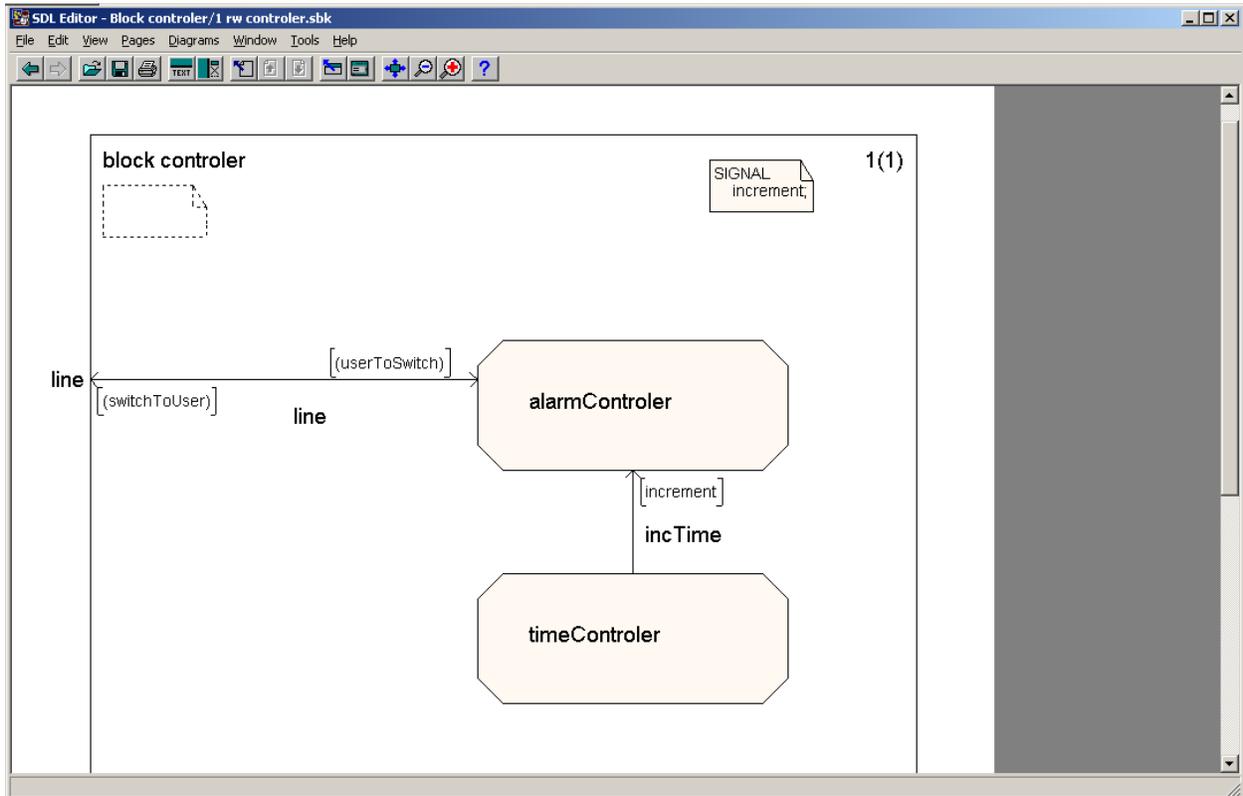
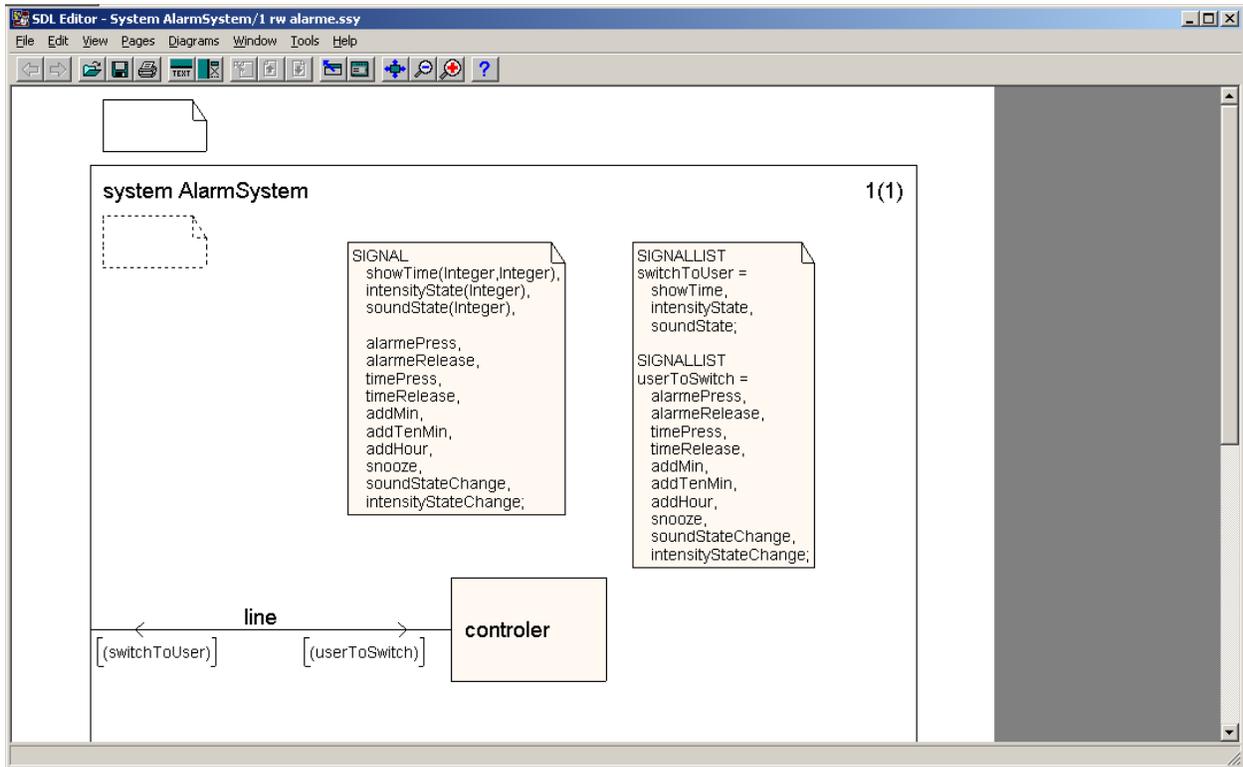
Le cas d'utilisation « alarme » décrit le cas types où l'alarme sonne. Le diagramme de séquence correspondant à ce cas d'utilisation est sauvegarder dans le fichier « UseCase4_5.msc ».

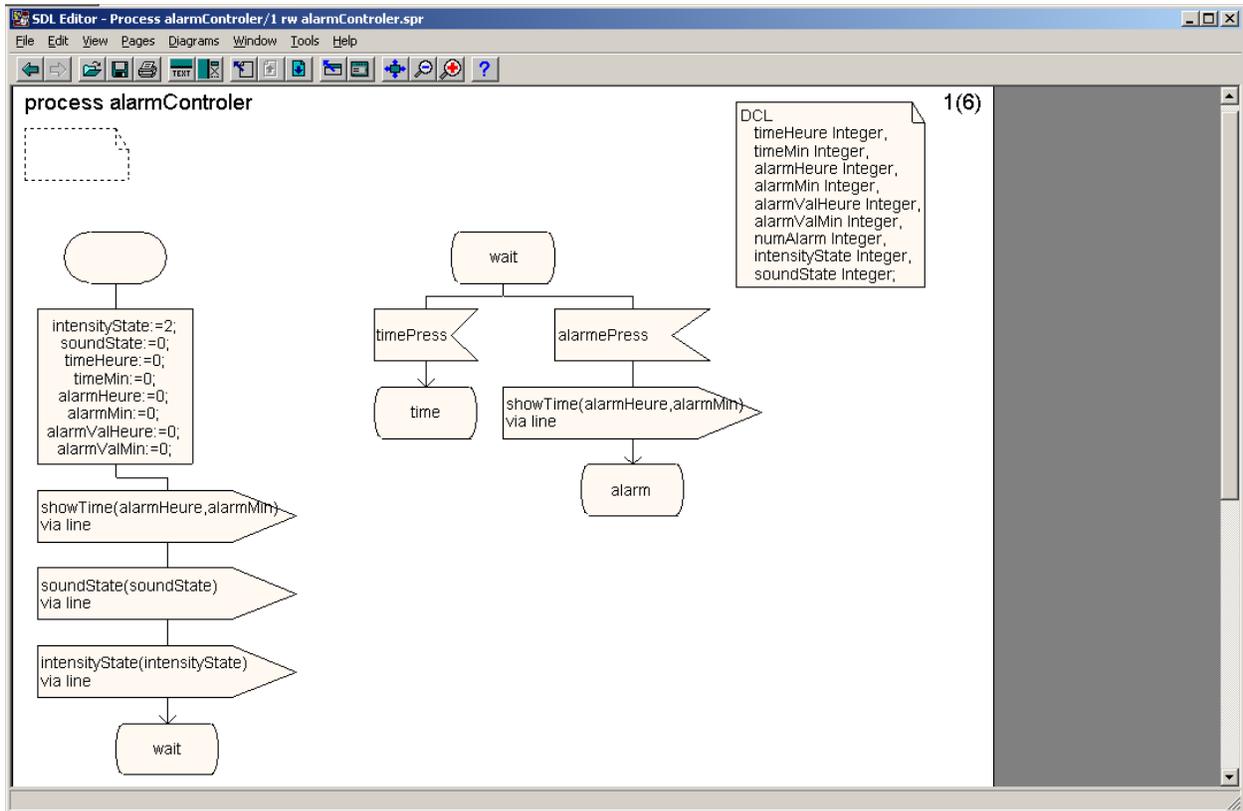
Le cas d'utilisation « snooze » décrit le cas types où l'alarme sonne et ensuite l'usager appuie le bouton « snooze ». Le diagramme de séquence correspondant à ce cas d'utilisation est sauvegarder dans le fichier « UseCase4_5.msc ». Après un initialisation, et avoir préciser l'heure de l'alarme, on remarque que le réveille-matin entre dans l'état beeping, ce qui indique que l'alarme sonne. L'usager peut peser « snooze ». On remarque que l'alarme sonne cinq minutes plus tard. Après avoir sonné pour une minute, l'intensité de l'alarme augmente. Après une deuxième minute, le réveille-matin lâche de sonner.

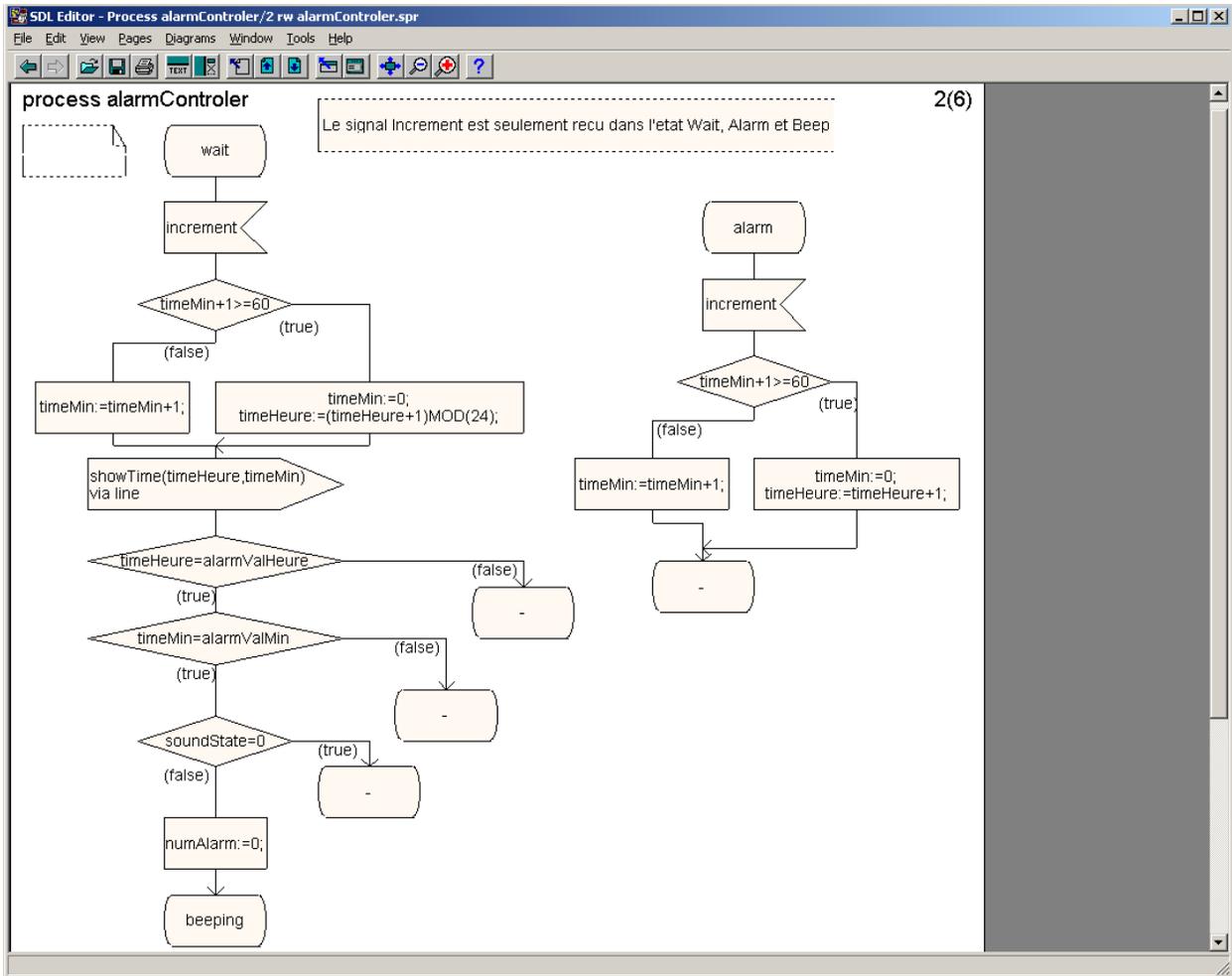
Systeme SDL :

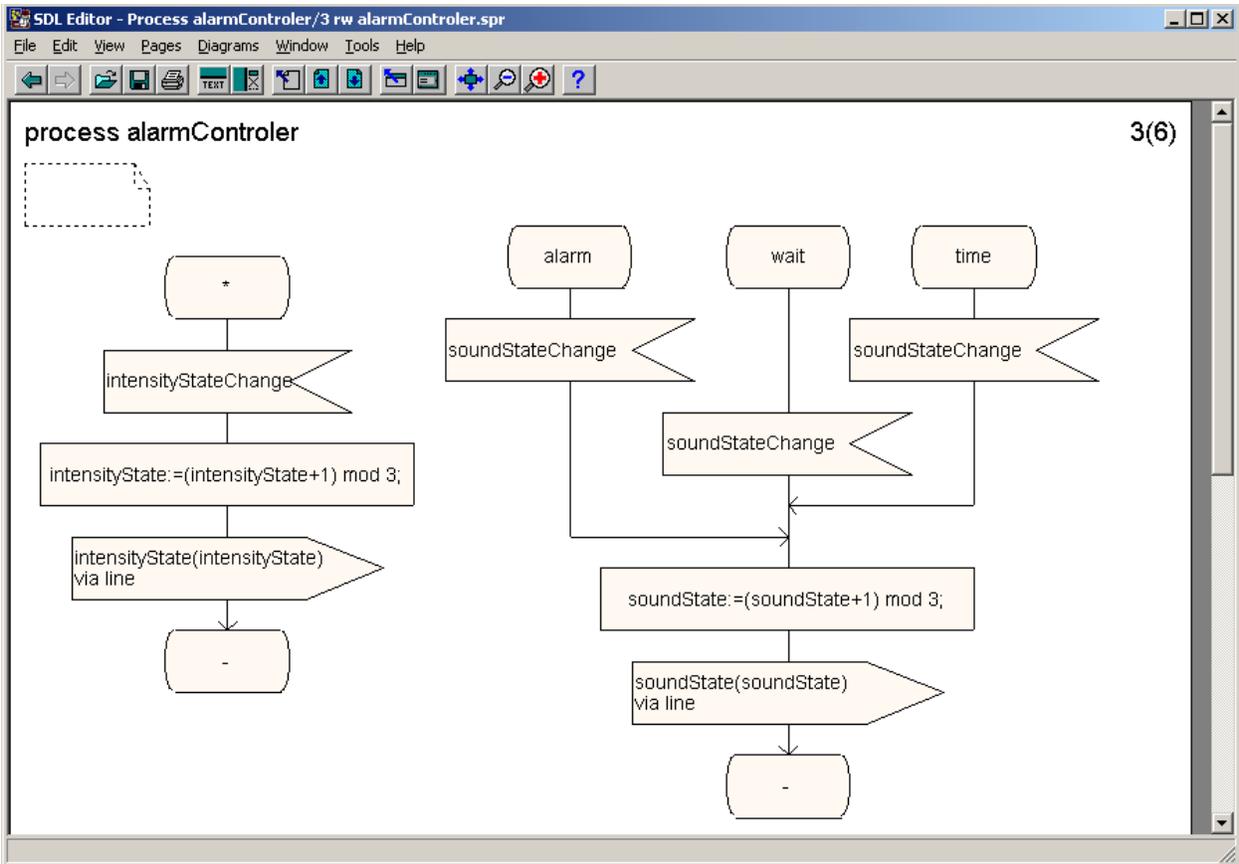
Voici le système de réveille-matin, comme décrit par les exigences, en SDL. Il se trouve sauvegarder dans le fichier devoir1.sdt

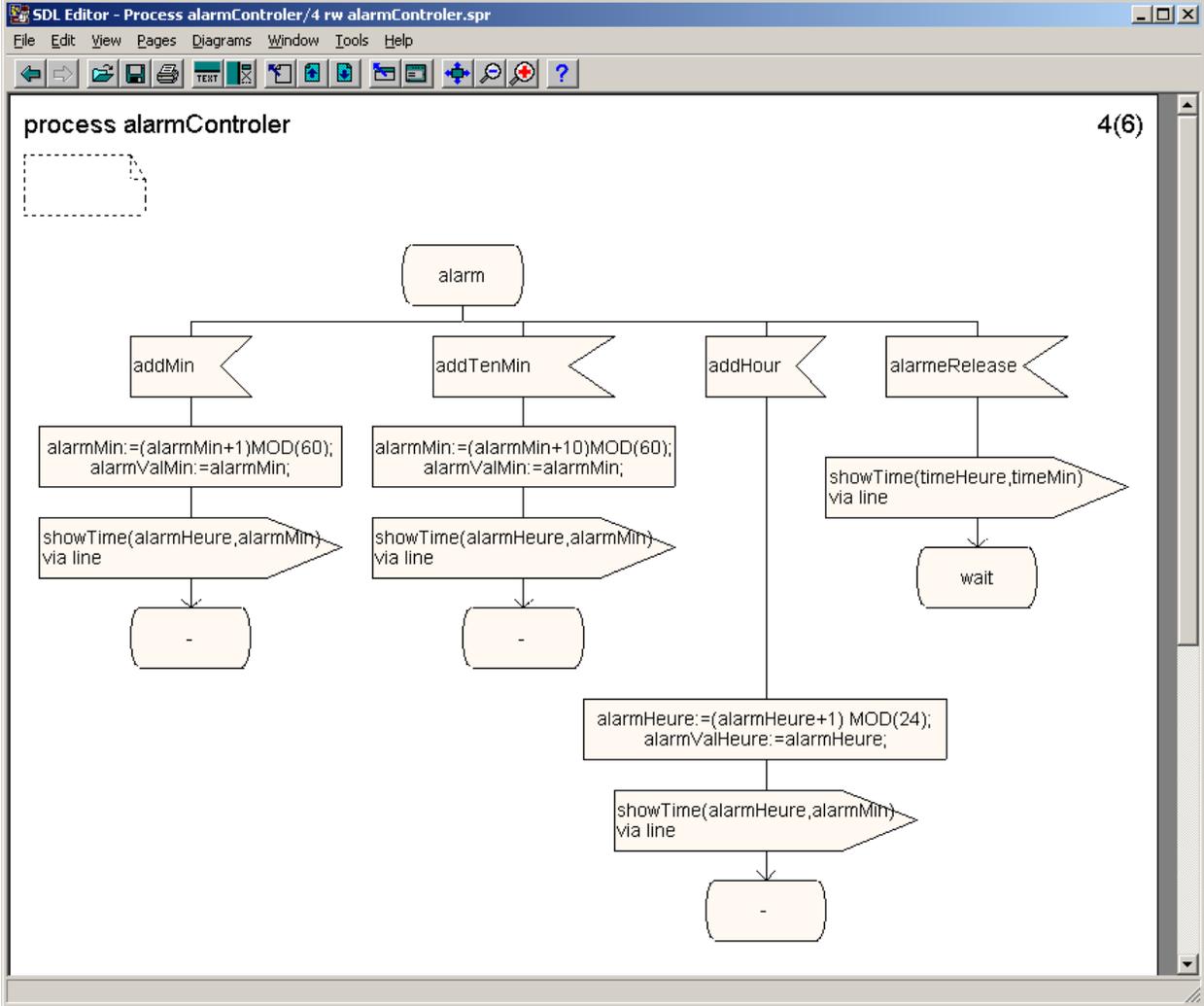


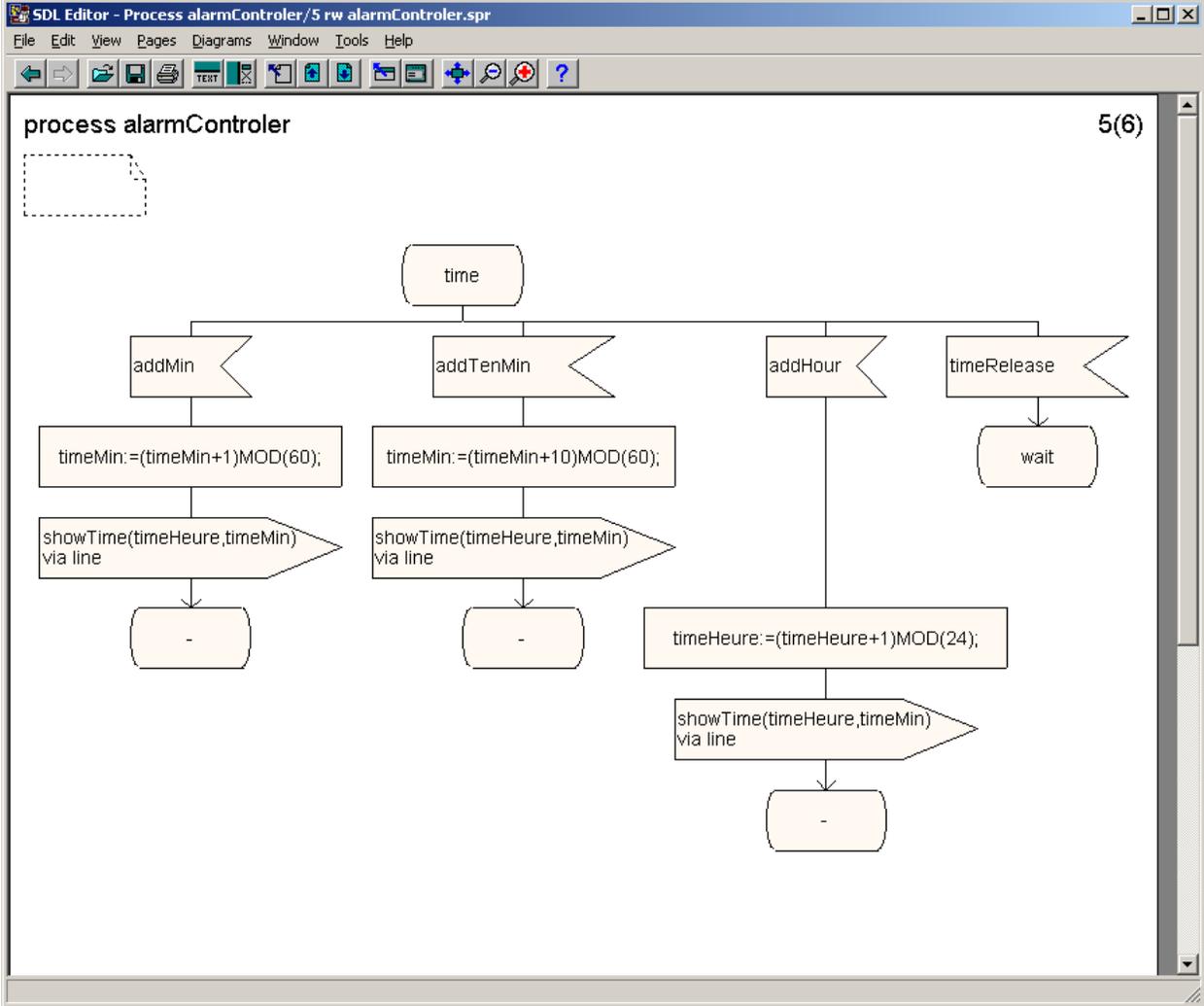


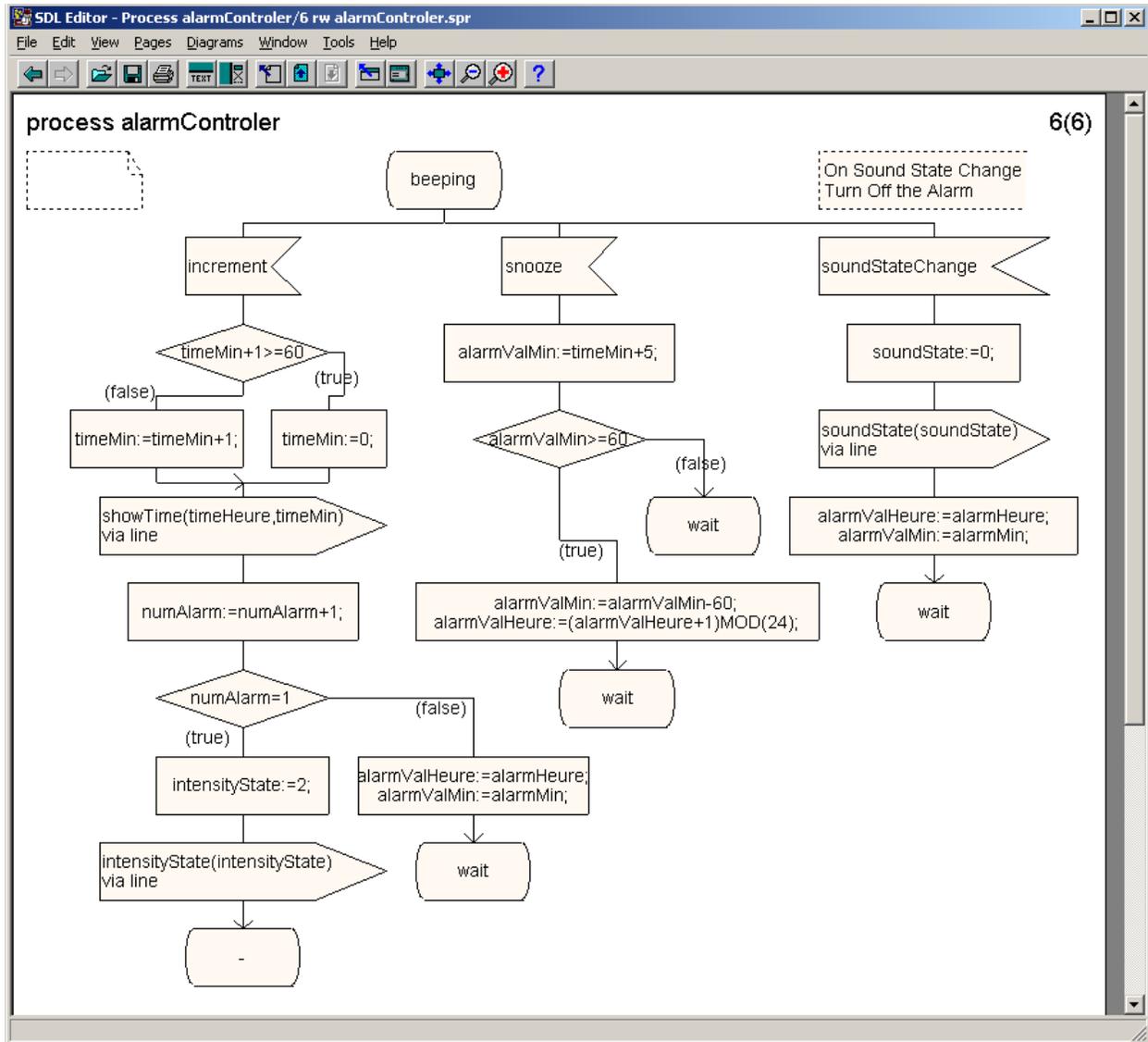












Discussion et Description des simulations exécutées :

Description des simulations exécutées: S.v.p. donner une courte description (une demie à une page de texte) des activités de vérification que vous avez faites, quelles erreurs ont été initialement détectées et ensuite corrigées; indiquez si la vérification a finalement été un succès, etc.

Plusieurs simulations ont été utilisées pour nous assurer de la validité de notre code. Lors de l'exécution du cadra, l'heure 0:00 s'affiche. Nous avons testé la modification de l'heure pour ajuster cette heure à l'heure actuelle et des cas extrêmes dont 23h59. Le cadran fait des cycles de 24 heures avec succès.

Par la suite, nous avons testé l'option « snooze » en configurant l'alarme à 23h58 et appuyant sur « snooze » pour tester si l'alarme sonnera 5 minutes plus tard à 00h03 et cela a fonctionné avec

succès. L'intensité se modifie après la première minute, l'alarme s'éteint après la 2^e minute et en tout temps l'utilisateur peut appuyer sur le bouton « SoundState » pour éteindre l'alarme.

Nous avons aussi pensé à garder un backup de l'heure de l'alarme pour que lorsque le bouton snooze est appuyé, l'heure de l'alarme ne soit pas modifiée pour la journée suivante et que l'alarme sonne 5 minutes plus tard.

Nous avons eu des petits problèmes sur les frontières dont le cadran allait de 0 à 58 minutes au lieu de 0 à 59 minutes ce qui n'a pas prit beaucoup de temps à corriger. Nos plus gros problèmes était lors de la conception, nous avons commencé en utilisant des variables TIME où nous nous sommes complètement perdu dut a de mauvaise connaissance de sont utilisation. Par la suite, nous sommes été avec une variable integer de 0 à 1440 pour représenté une journée. La fonction modulo est par conséquent restreinte de 0 à 256 et ne fonctionne pas pour les valeurs plus grandes. Finalement, nous avons modifié notre code en utilisant 2 variables pour représenter les heures et les minutes.