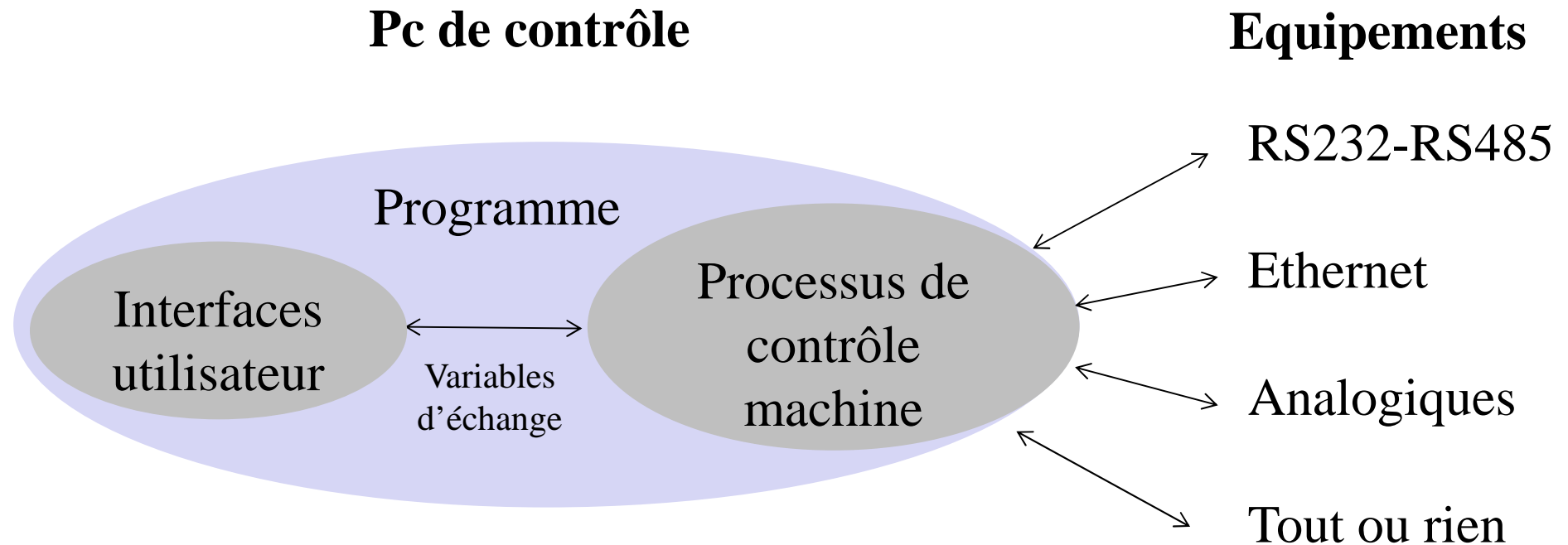
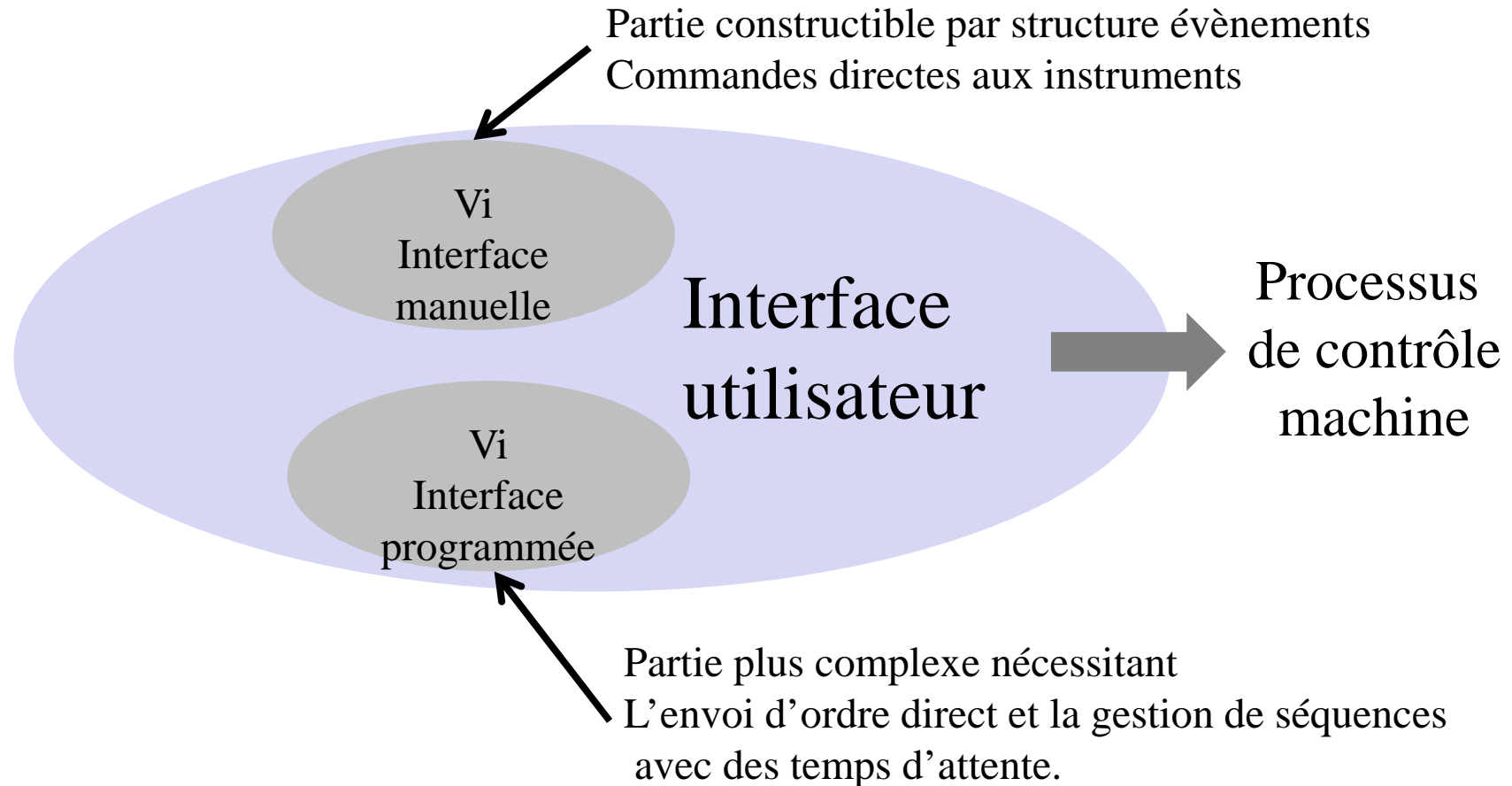


# Structure générale d'un programme de supervision Géré par boucle d'évènement



## L'interface Utilisateur



**Problème: Ne pas bloquer l'interface durant l'exécution de séquence avec des délais long**

# Exécution de séquences longues dans un programme à structure d'évènements sans blocage

## Problématique

Dans les programmes de contrôle/commandes communiquant avec plusieurs équipements le timing entre les évènements (communications, actions utilisateur, programmes) est fondamental.

Pour optimiser le temps machine d'un programme entre ses besoins et les requêtes extérieures, on utilise des structures d'évènements.

**Structure d'évènement** —————> **Système par interruptions logiciels**

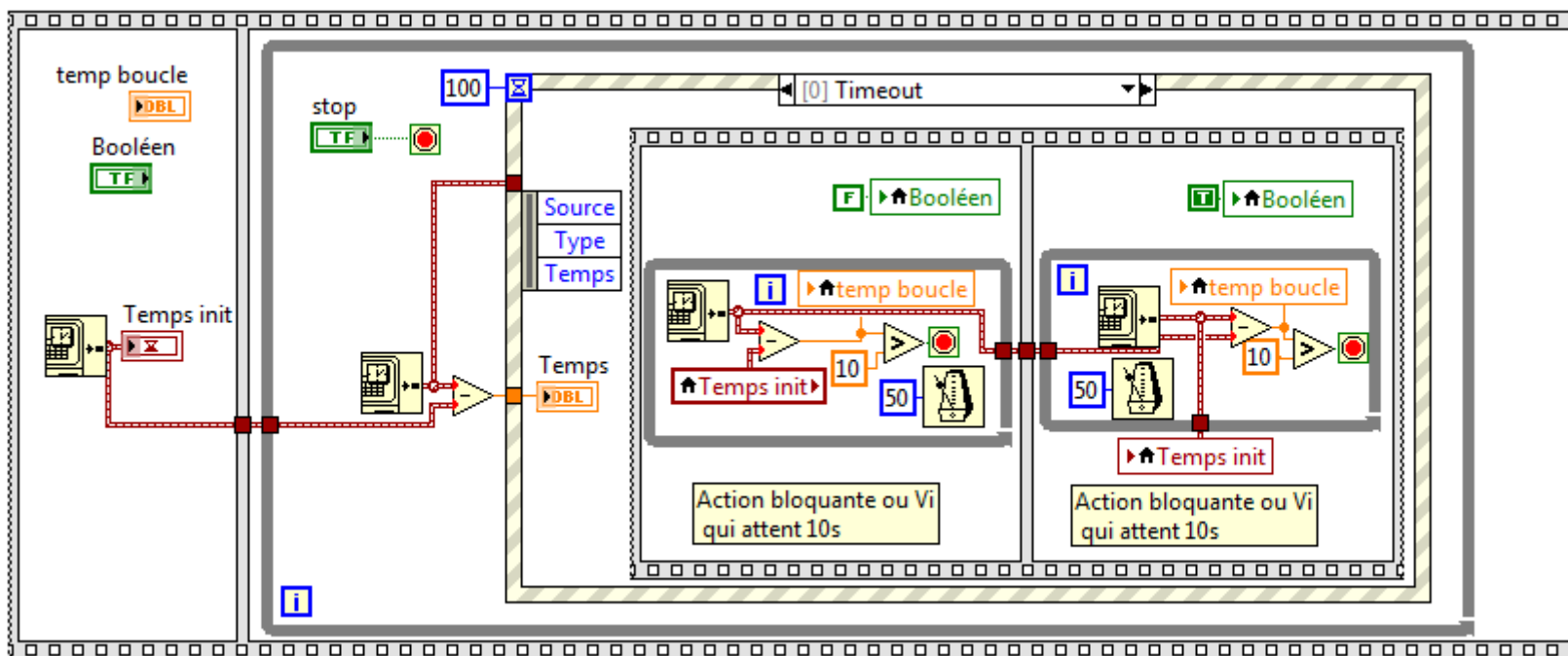
-Une action qui met en pose le programme en cours  
Et déclenche le déroulement d'une fonction prioritaire.

La fonction doit être courte pour ne pas bloquer la structure.

**Impossible d'utiliser un délai long dans la structure sans faire de bugs dans le programme.**

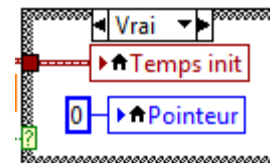
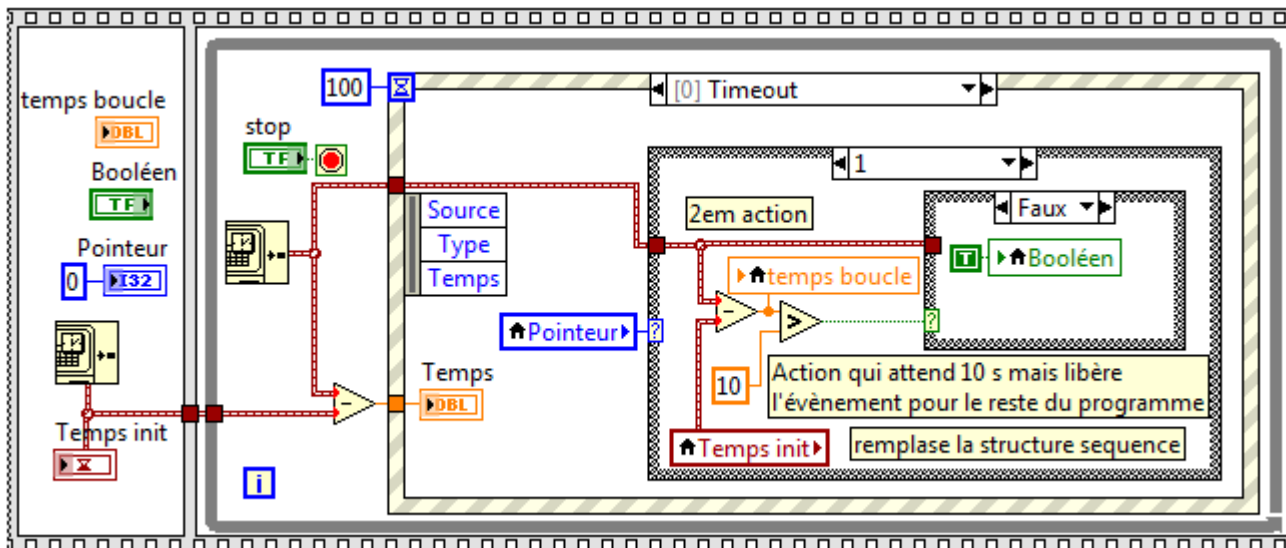
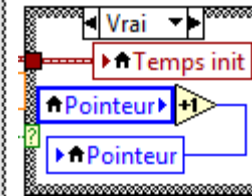
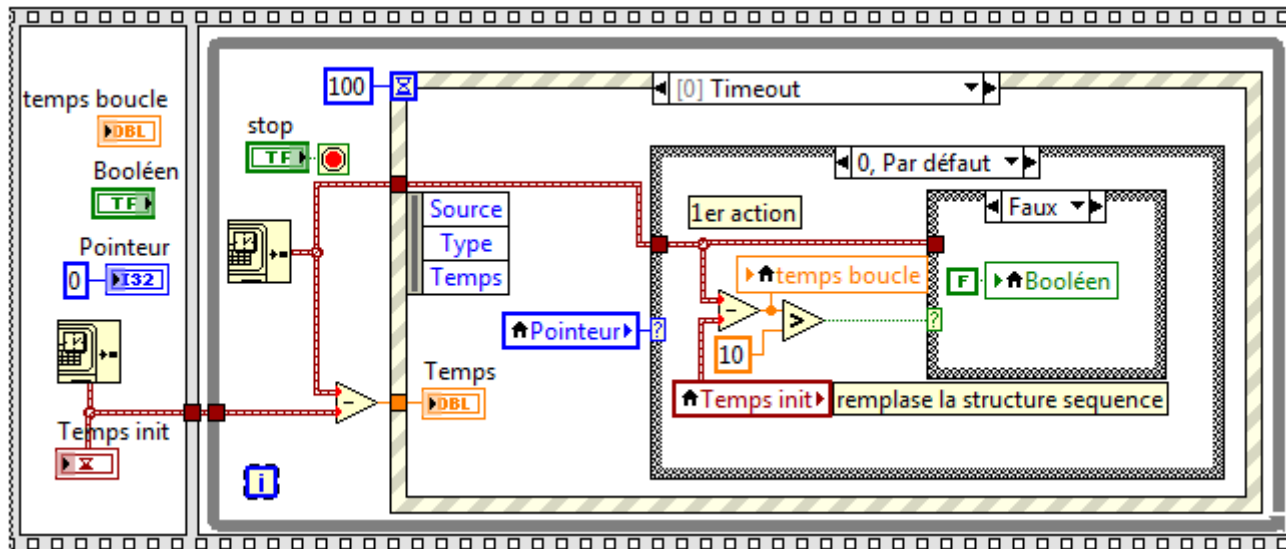
## Exemple de Vi bloquant

Cet exemple ne peut prendre en compte le bouton stop que toutes les 20s, et la variable Temps n'est pas actualisée



**Blocage de la structure dans l'évènement « timeout »**

# Exemple de Vi Non bloquant



## Conclusion

- Technique qui permet de faire du séquentiel dans un système combinatoire
- Facile à écrire sous LabView.
- Il est possible d'imbriquer des étapes et de faire des sauts entre étapes.
- précaution      —→ Bonne gestion du pointeur d'étape  
mettre à jour après la fin de chaque étape..