

TD 12 : Tests de validation et vérification

Génie Logiciel Avancé – M1 II

15–16 avril 2010

1 Carte météo

Un système de *carte météo* (CM) doit générer des cartes avec la météo de chaque région en se basant sur les données collectées à partir des *stations météo* (SM) ou autres sources comme les ballons-sondes ou satellites. Les SM transmettent leurs données à un central de région (CR) en réponse à une requête de celui-ci.

Un CR valide les données collectées des SM et les intègre avec les données des autres sources. Les données intégrées sont archivées. En utilisant les données de cette archive et une BD de cartes numériques, le CR génère toutes les heures un ensemble de cartes météo locales. Ces cartes peuvent être imprimées ou affichées en différents formats.

La figure 1 décrit l'architecture logiciel de haut niveau du système CM.

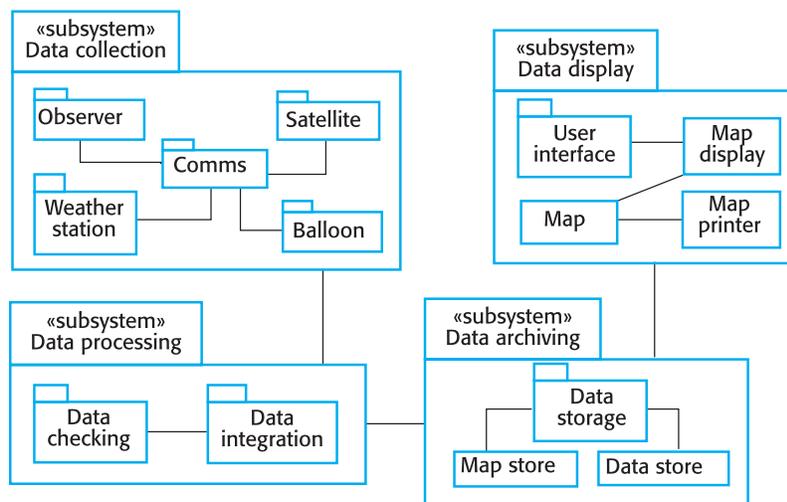


FIG. 1 – Architecture logiciel d'un système de carte météo.

Exercice 1:

Plan de test

En utilisant l'architecture de ce système, constituez un plan de test en précisant plus particulièrement :

1. les phases du test à prévoir,
2. le plan du test d'acceptation pour les utilisateurs,

3. les composantes à tester et
4. les besoins en matériel et logiciel à prévoir.

Exercice 2:

Test de système : test d'acceptation

En utilisant l'architecture donnée pour l'interface utilisateur, proposez une suite de tests à effectuer pour démontrer que le système satisfait ses charges.

Exercice 3:

Test de système : test d'intégration

Proposez une procédure *top-down* et une procédure *bottom-up* pour effectuer le test d'intégration des composantes du système CM.

Exercice 4:

Test de système : test de performance

Proposez une suite de tests qui montrent les performances en temps et au limites du système CM.

2 Station météo

Dans la suite, nous allons nous intéresser plus particulièrement au sous-système station météo (SM). Une SM est un ensemble d'instruments de mesure contrôlés par un logiciel. Le logiciel collecte les mesures et l'état des instruments, effectue des calculs sur les données et transmet ces données au CR. Les instruments inclus dans un SM sont : deux thermomètres un de l'air et l'autre du sol, un anémomètre¹, une manche à vent, un baromètre et un pluviomètre. Les données sont collectées périodiquement. Quand une commande de collecte arrive, le SM fait un rapport des données collectées et le transmet au CR.

La figure 2 décrit les architectures de haut niveau et détaillée du sous-système SM.

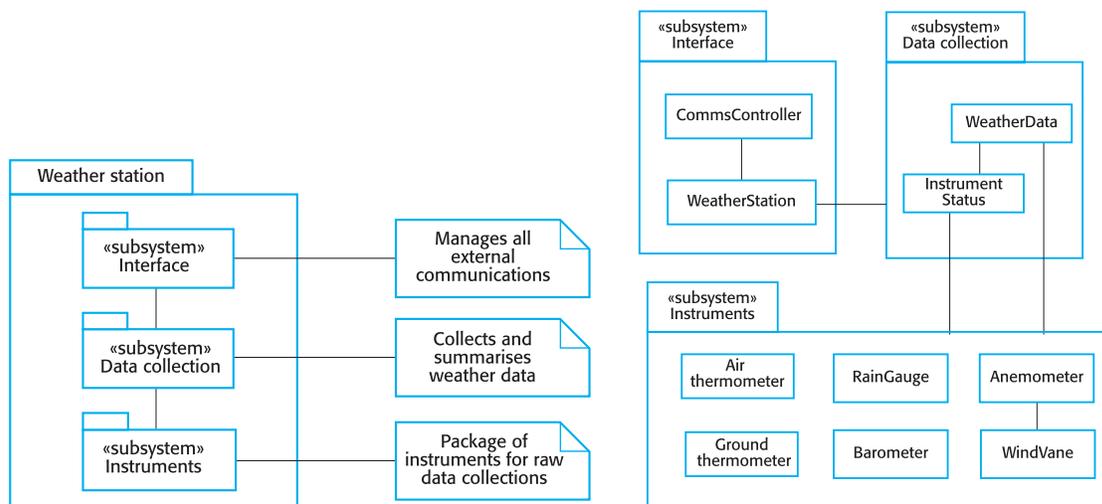


FIG. 2 – Architecture d'un sous-système station météo.

¹Appareil permettant de mesurer la vitesse ou la pression du vent (anemos en grec).

Les cas d'utilisation de ce sous-système ainsi que le diagramme de flot des services du SM sont donnés dans la figure 3.

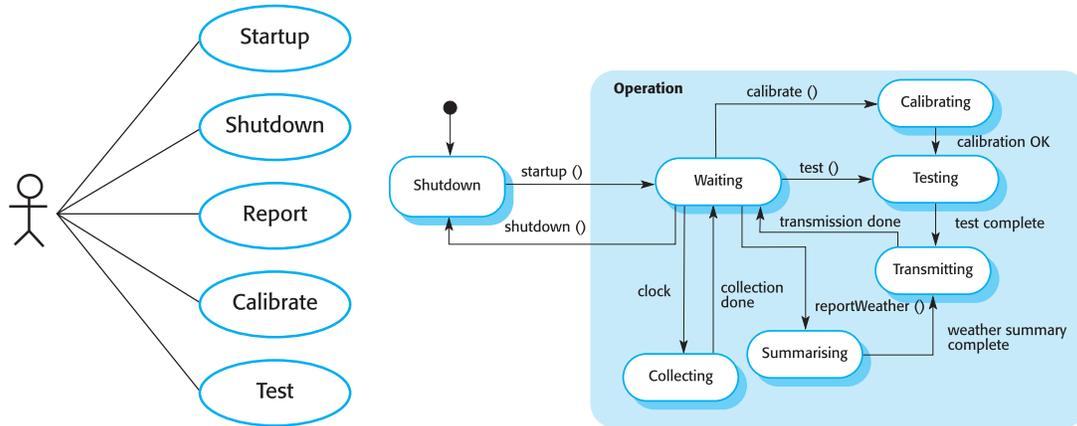


FIG. 3 – Le diagramme d'utilisation et le diagramme de flot pour la SM.

Exercice 5:

Test de système : test de fonctionnement

En utilisant le diagramme de flot, proposez une suite de tests pour valider le fonctionnement correct du sous-système SM.

Dans le cahier de charge, le cas d'utilisation **Report** est spécifié à l'aide de la fiche ci-dessous et du diagramme de séquence (pour l'utilisation nominale) de la figure 4.

Système : Station météo

Cas d'utilisation : Report

Acteurs : Sous-système de collecte de données, Sous-système station météo

Description : La SM envoie un résumé des données collectées en utilisant ses différents instruments dans la période de temps observée depuis la dernière demande. Les données envoyées sont la température minimum, maximum et moyenne de l'air et du sol, la pression maximum et minimum de l'air, la vitesse minimum et maximum du vent, le volume total de pluie et les directions du vent mesurées toutes les 5 minutes.

Activation : Le sous-système de collecte de données établit une connexion modem avec le système de communication et demande la collecte de données.

Effet : Le résumé des données collectées est transmis au demandeur.

Commentaires : Les SM sont interrogées usuellement avec une fréquence de 1 fois toutes les heures mais cette fréquence peut être différente d'une SM à autre et peut augmenter dans le futur.

Exercice 6:

Test de composante : en boîte noire

En utilisant le diagramme de flot et le diagramme de classe, proposez une suite de tests pour tester la classe `WeatherStation`. Pourquoi s'agit-il de tests en boîte noire ?

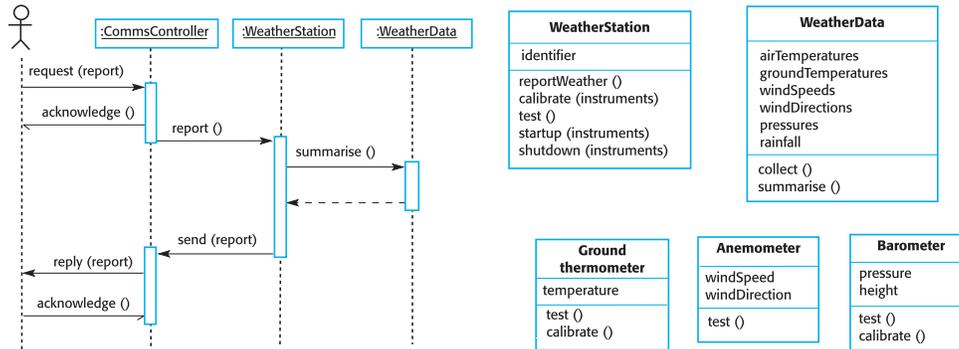


FIG. 4 – Diagramme de séquence et diagramme de classe pour la SM.

Exercice 7:

Test de composante : fonction Report

En utilisant le diagramme de séquence, proposez une suite de tests pour tester le cas d'utilisation Report.

Exercice 8:

Test de composante : partitionnement des entrées

La méthode summarize() de la classe WeatherData utilise une méthode computeAv spécifiée comme suit :

```

/**
 * Compute the average value of the sequence values.
 * Pre-condition: values is not empty, values.length>=1
 * Post-condition:
 *   - the average can be computed and the return is the value of the average
 *   or - the average cannot be computed and the result is the float constant NaN
 */
Float computeAv(Float[] values)
  
```

Donnez une partition de l'ensemble des entrées de cette méthode qui regroupe les valeurs des entrées qui génèrent les même type de sorties. Pour chaque partition identifiée, écrivez une suite de cas de tests en précisant les données de test et les résultats attendus. Quel type de méthode de test (boîte noire ou blanche) utilise le partitionnement des entrées ?

Exercice 9:

Test de composante : test structurel

1. Donnez une implémentation pour la méthode computeAv ci-dessus.
2. En partant du code écrit, construisez le graphe de contrôle (CFG) de la méthode.
3. Ecrivez une suite de tests qui couvre toutes les instructions du CFG.
4. Ecrivez une suite de tests qui couvre toutes les décisions du CFG.
5. Ecrivez une suite de tests qui couvre toutes les combinaisons décision et condition atomique (MCDC) du CFG.