ETUDE DE CAS UML : SYSTEME DE PLANIFICATION

C. FORCE

www.cetus-links.org

Synthèse de la spécification des besoins

- Les professeurs doivent surveiller les écoliers pendant les récréations dans diverses parties d'un établissement.
- L'affectation des professeurs à chaque récréation et à chaque lieu est donnée par le "planning des récréations".
- Les professeurs sont affectés aux récréations proportionnellement à leur temps d'enseignement.
- Les professeurs peuvent préciser des périodes où ils ne peuvent pas surveiller les récréations.

Scénario principal : construction d'un planning

- Précondition : un planning non terminé est affiché.
- Scénario nominal :
  - 1 - L'utilisateur dispose d'une liste des professeurs et la liste des impossibilités de chacun.
  - 2 - Par glisser déplacer les noms des professeurs sont placés sur les cases vides du planning, jusqu'à ce que chaque case soit occupée par un et un seul professeur.
  - 3- L'utilisateur pourra faire plusieurs essais de placement (enlever, déplacer, ajouter des noms dans les cases).
  - 4 - Chaque fois qu'un professeur est affecté à une récréation, les résultats synthétiques sont mis à jour.
- Postcondition : une nouvelle version du planning est enregistrée

Utilisateurs

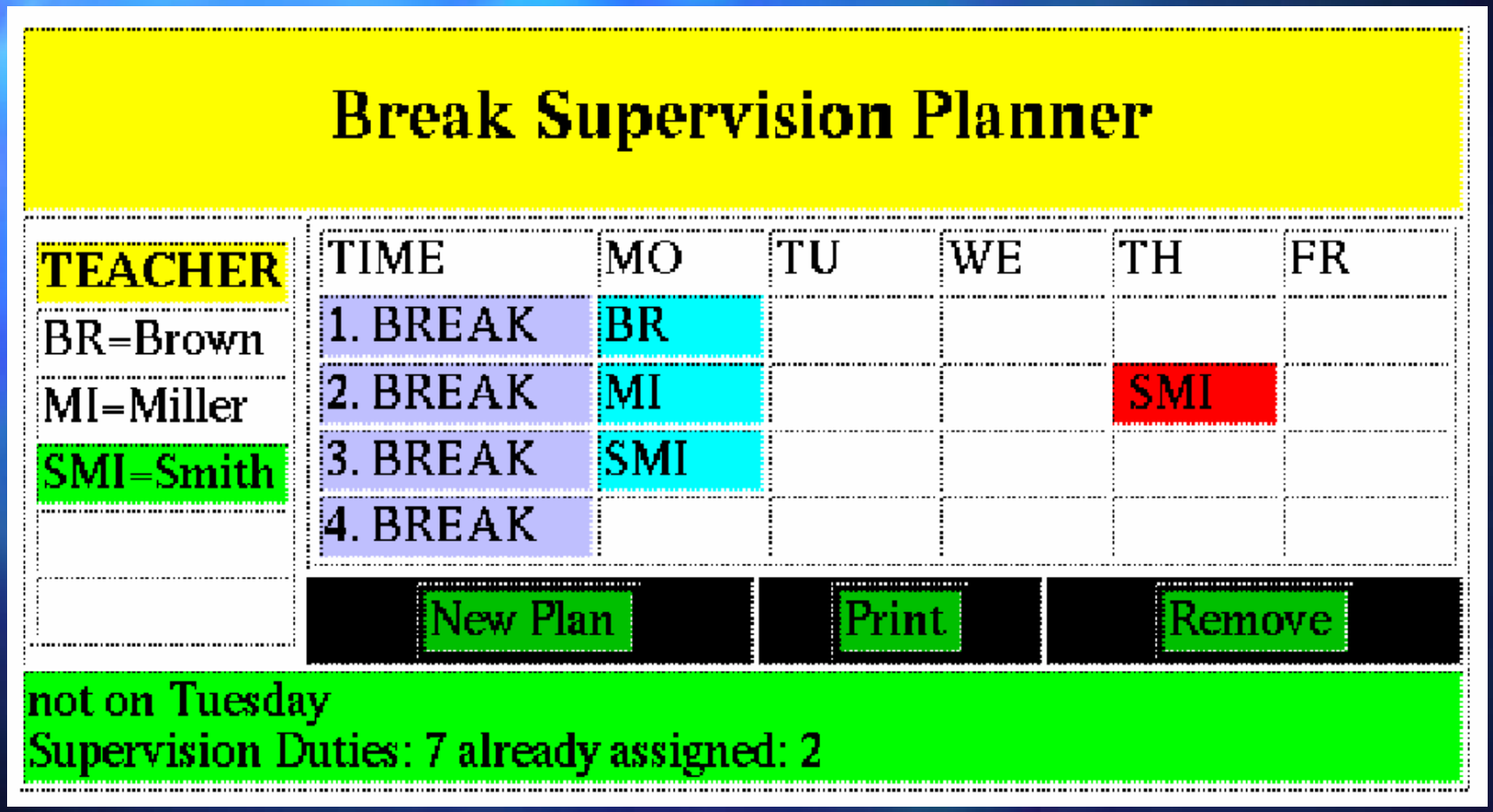
- Les responsables de l'édition des plannings de récréations (appelés planificateurs),
- Les gestionnaires des données concernant l'équipe d'enseignement (service du personnel).

Besoins non fonctionnels

- Utilisation multipostes,
- Gestion de données persistantes (fichiers…),
- Autodescription,
- Système cible : PC/Windows ou Unix.

Maquette de l'interface du système

Figure 1

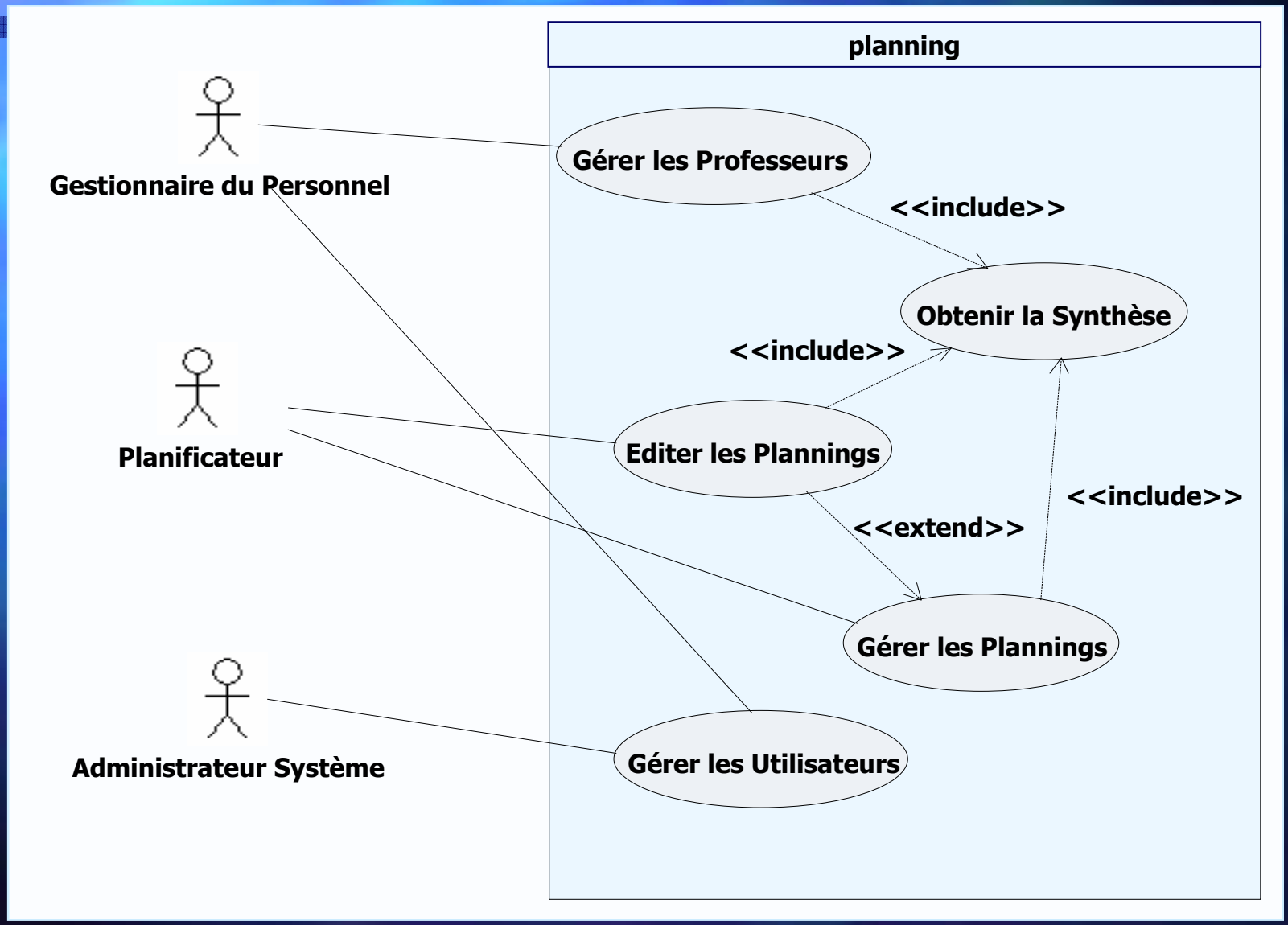


Spécification des Exigences et Analyse

- Le modèle des cas d'utilisation est généralement facile à comprendre pour les utilisateurs,
- Les diagrammes de séquences et les diagrammes d'activités permettent d'exécuter des simulations pas à pas de la dynamique du système,
- Le modèle de classes du domaine montre les éléments d'information identifiés dans le modèle des cas d'utilisation,
- Les aspects dynamiques non triviaux sont spécifiés ensuite grâce aux diagrammes d'états des classes.
- Les opérations du système sont simplement décrites dans le dictionnaire de données, puisqu'elles n'ont pas de pré- et post-conditions complexes

Diagramme des Cas d'Utilisation

Figure 2



Description des cas d'utilisation

- ***Gérer les Plannings***
  - Manipuler les plannings comme un tout, comprend la création, la suppression, l'ouverture, la fermeture d'un planning et l'impression de tout un planning.
  - Fréquence : 1 fois par semaine à 1 fois par jour.
  - Données : Planning.
  - Sécurité : moyenne.
- Etc.

Description des utilisateurs

- ***Planificateur***
  - Le système peut être utilisé par les employés d'une même école (éventuellement des professeurs).
  - Plusieurs personnes peuvent accéder aux plannings à partir de différents postes de travail au même moment.
  - Un même planning ne peut être modifié que par une seule personne à un instant donné.
  - Tous les planificateurs ont les mêmes droits de modification.
- Nombre : une dizaine.
- Expérience : utilisateurs novices à expérimentés.
- Lieu : normalement à l'intérieur de l'école, accès par l'Internet possible.

Diagramme de Séquences : Editer les plannings :: Session d'édition

Figure 3

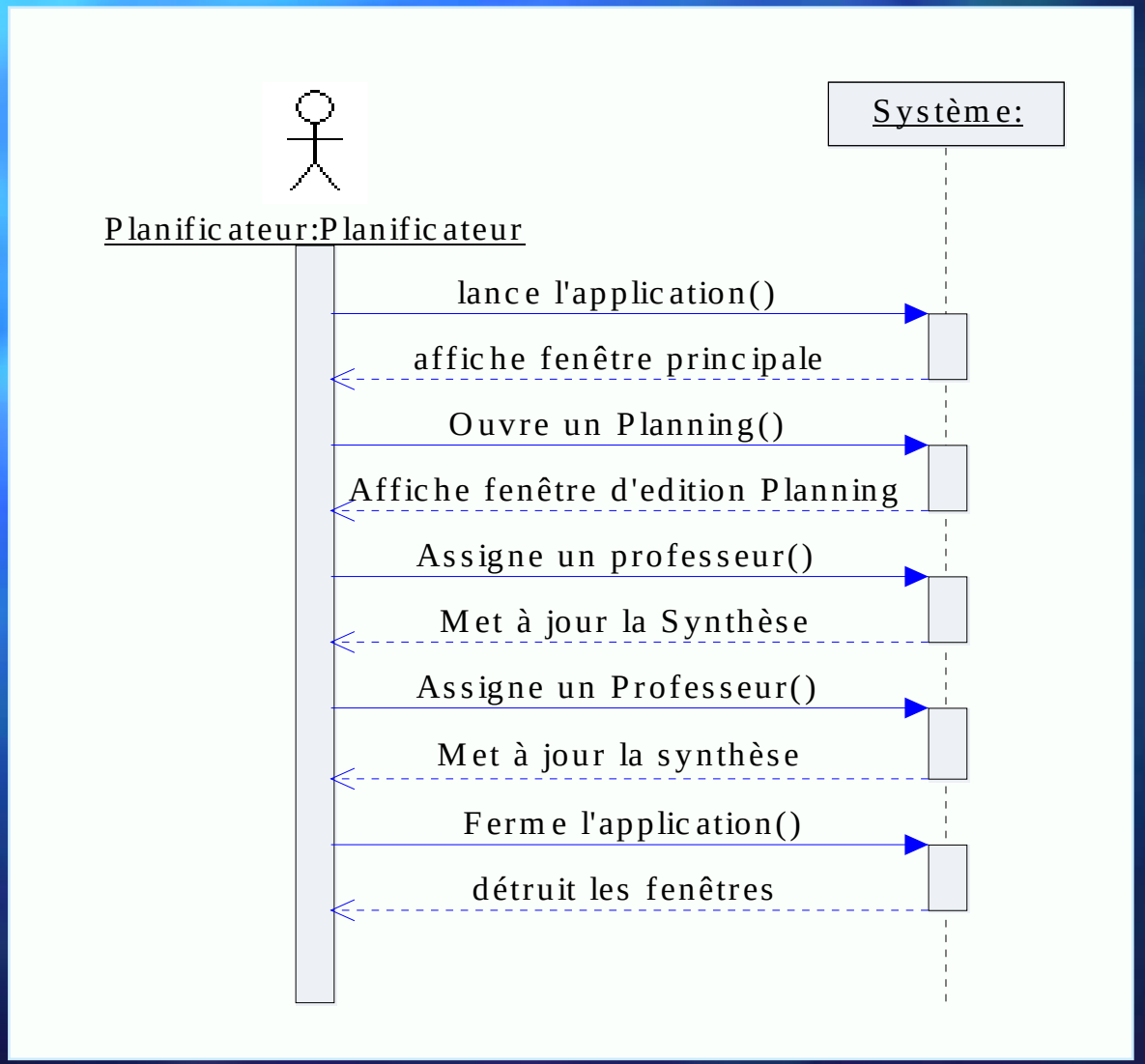
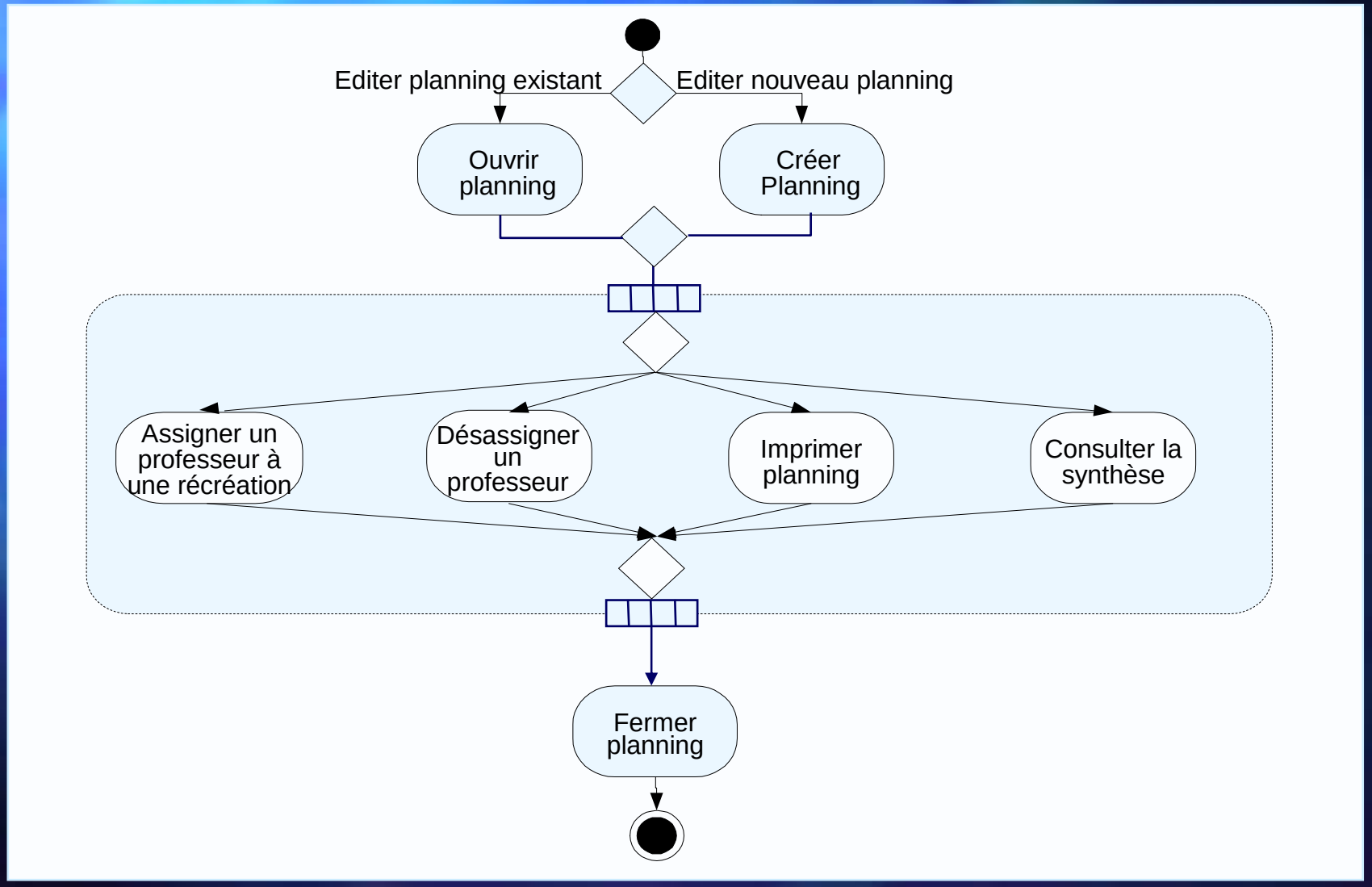


Diagramme d'Activités : Editer plannings

Figure 4



Cas d'utilisation : OBTENIR LA SYNTHESE

- ***Mise à jour à la demande*** : les utilisateurs doivent solliciter un bouton de mise à jour chaque fois qu'ils veulent voir la fenêtre de synthèse courante.
- ***Mise à jour périodique*** : la fenêtre des résultats de synthèse est mise à jour à intervalles de temps réguliers.
- ***Mise à jour à chaque changement*** : la fenêtre des résultats de synthèse est mise à jour dès qu'une modification se produit dans un planning.

Diagramme de séquences du Cas d'Utilisation Obtenir La Synthèse

(1) Mise à jour à la demande

Figure 5

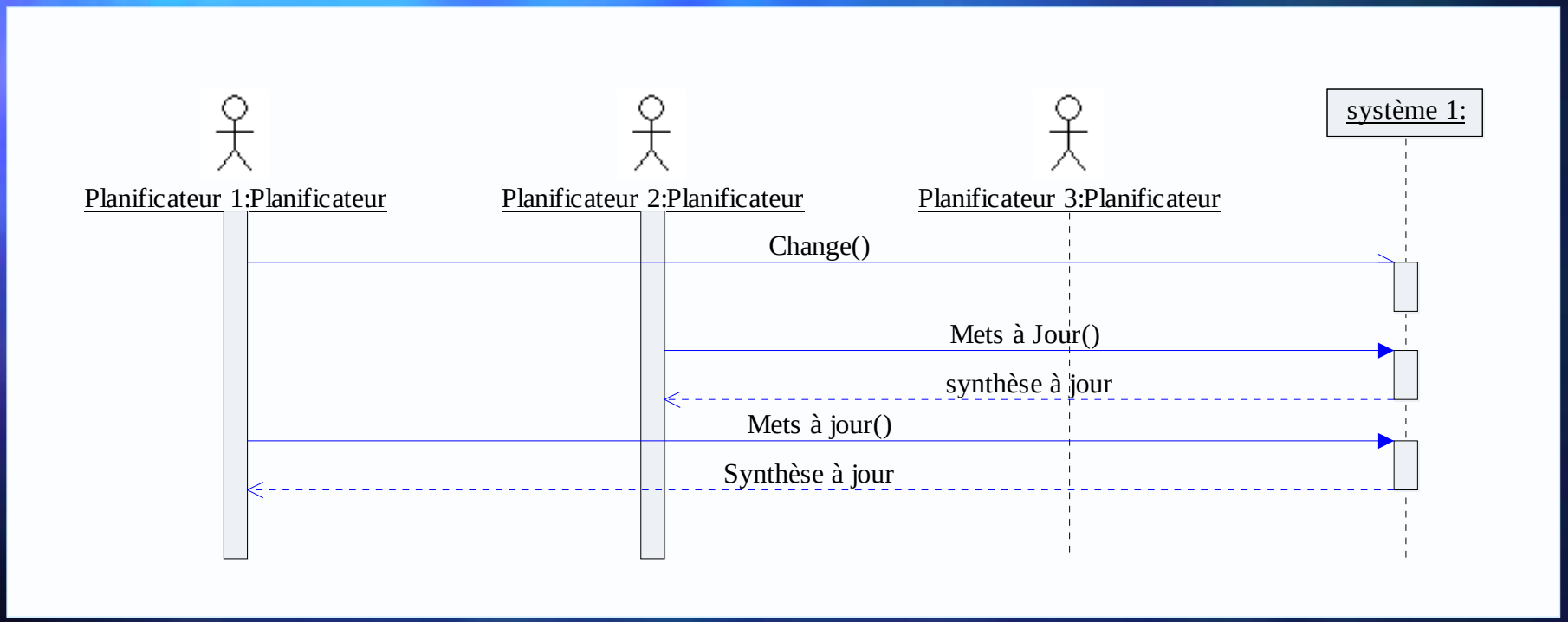


Diagramme de séquences du Cas d'Utilisation Obtenir La Synthèse

(2) Mise à jour à intervalles réguliers

Figure 5

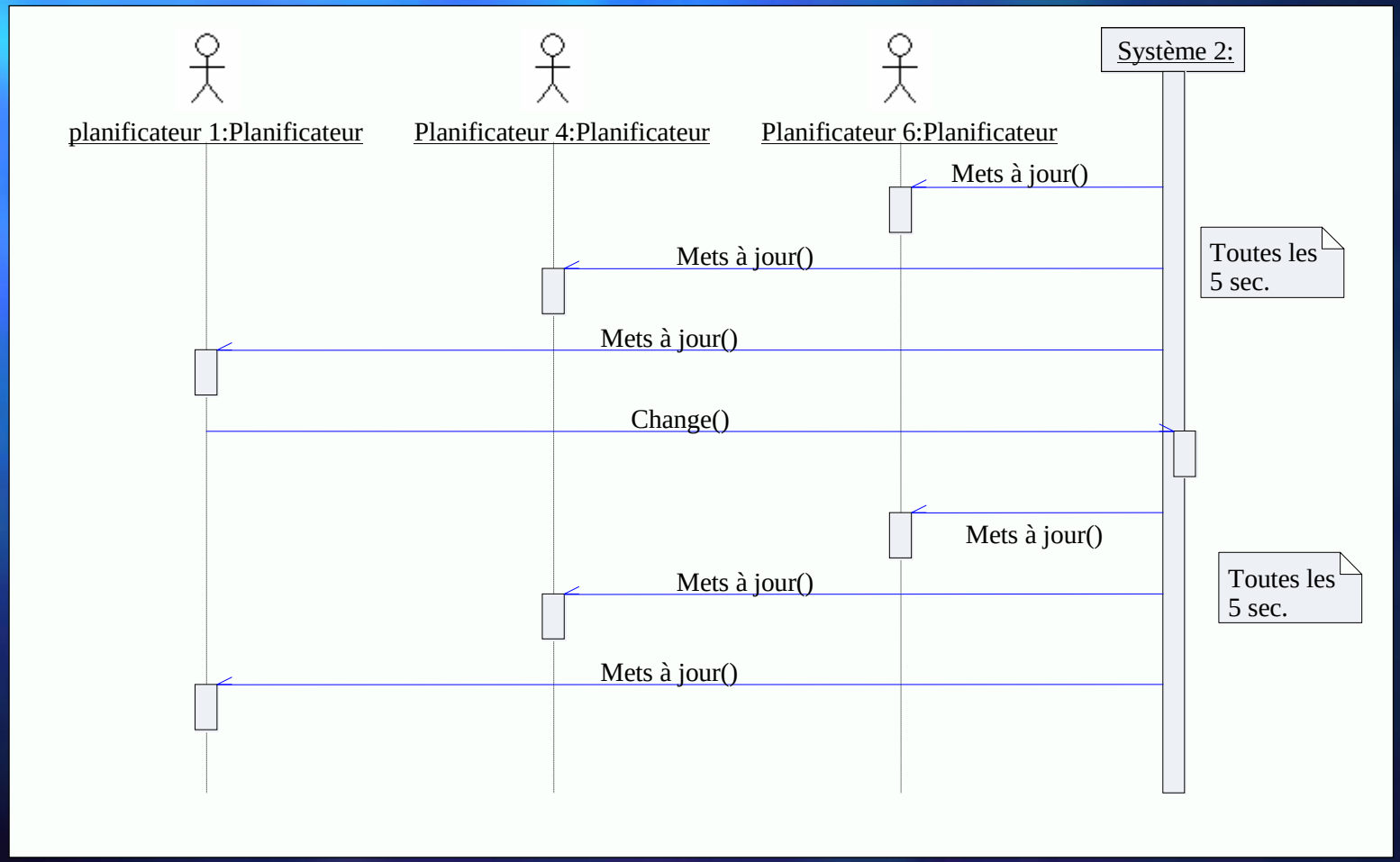


Diagramme de séquences du Cas d'Utilisation Obtenir La Synthèse

(3) Mise à jour à chaque changement - Figure 5

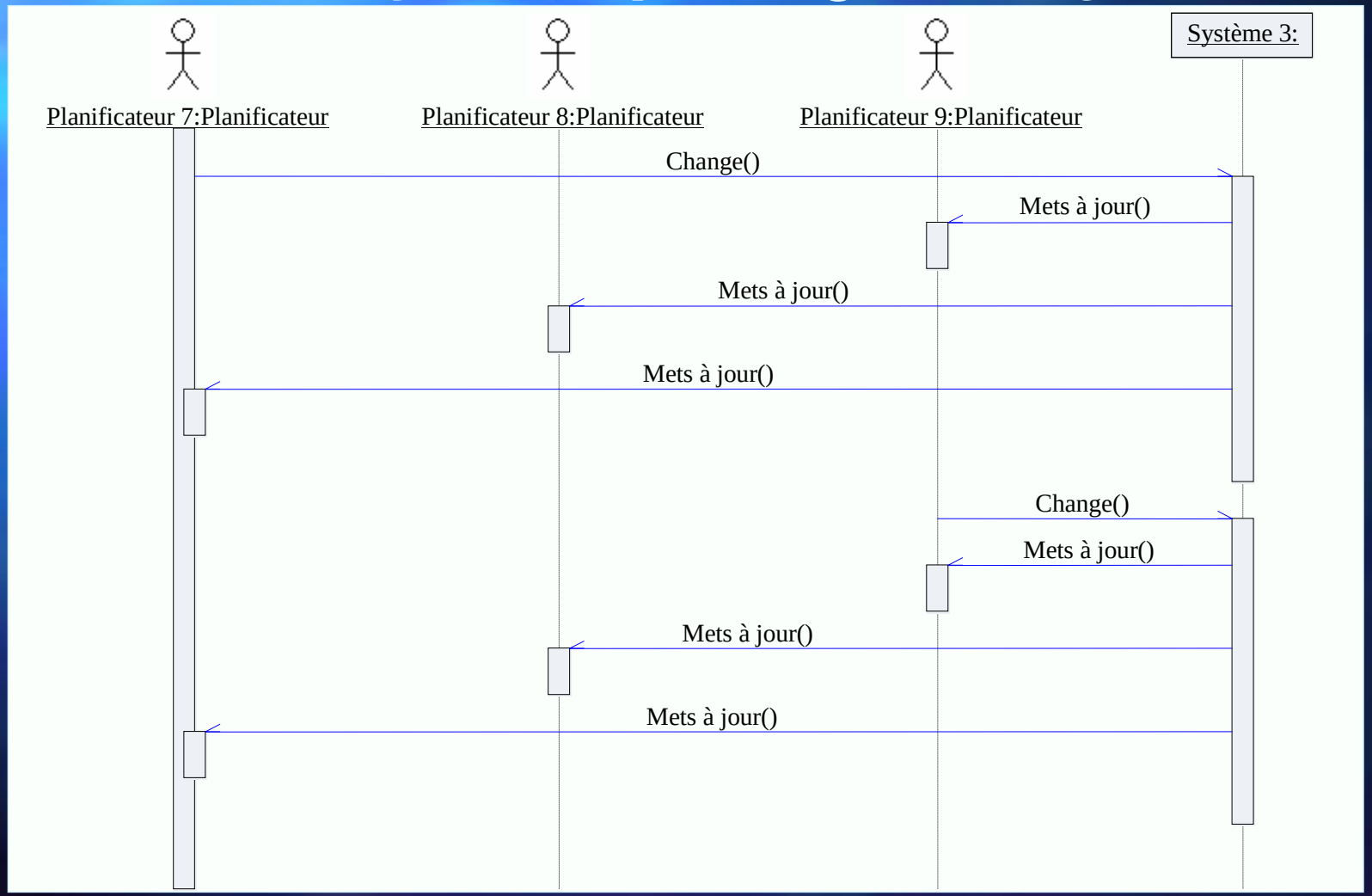
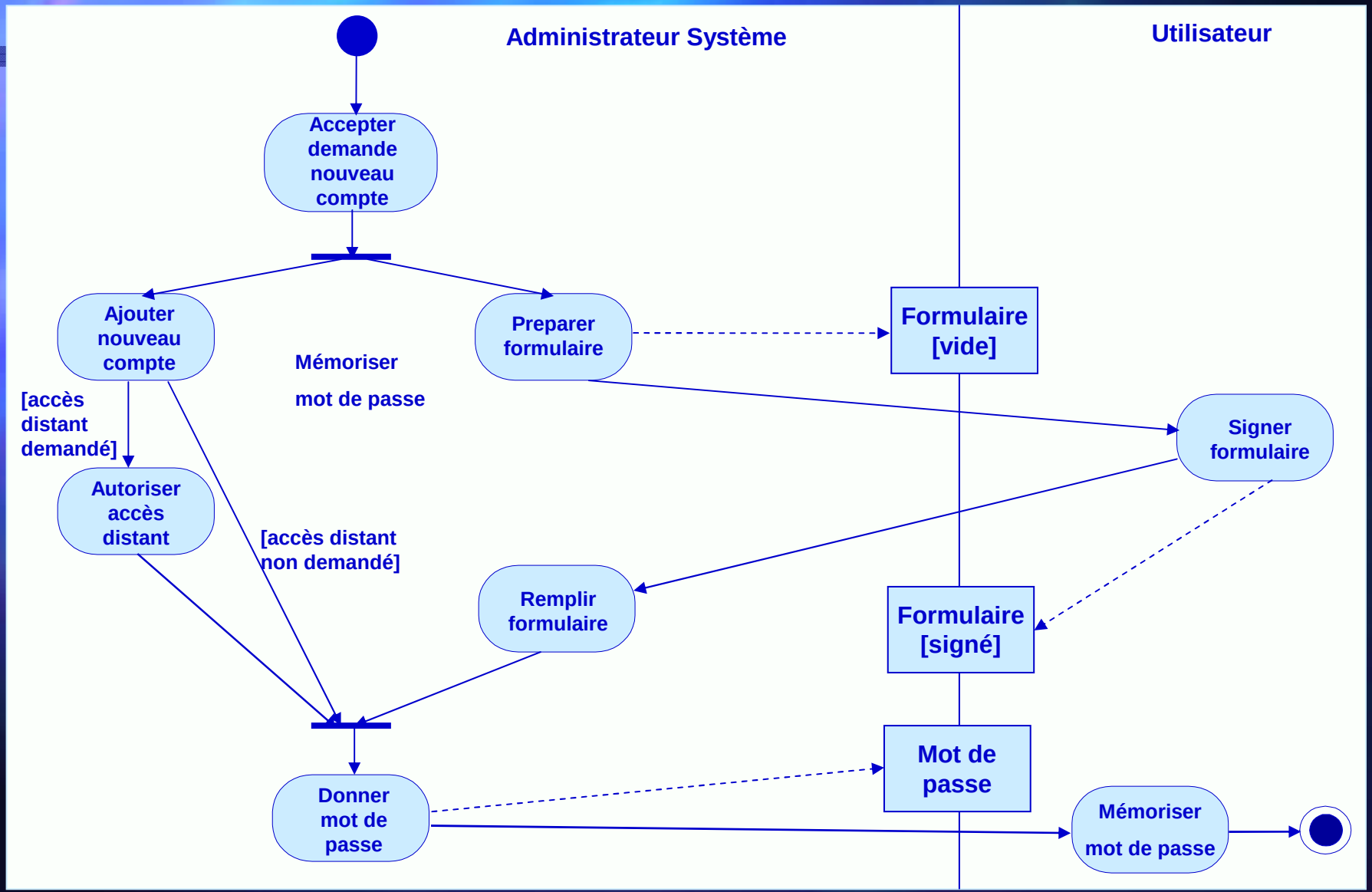


Diagramme d'activités GERER LES UTILISATEURS :: AJOUTER UN UTILISATEUR - Figure 6



Prototype de l'interface du système

Figure 7

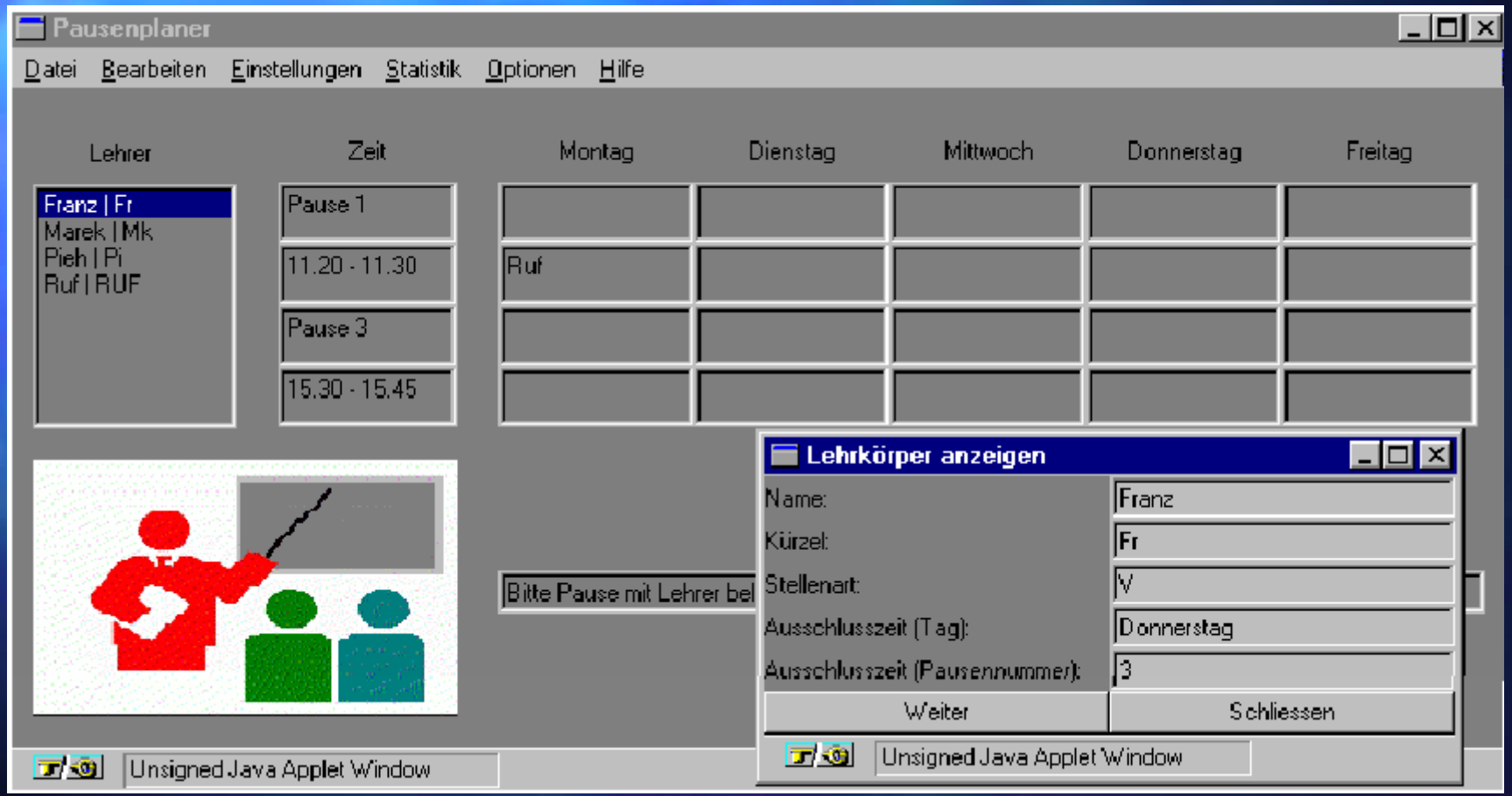
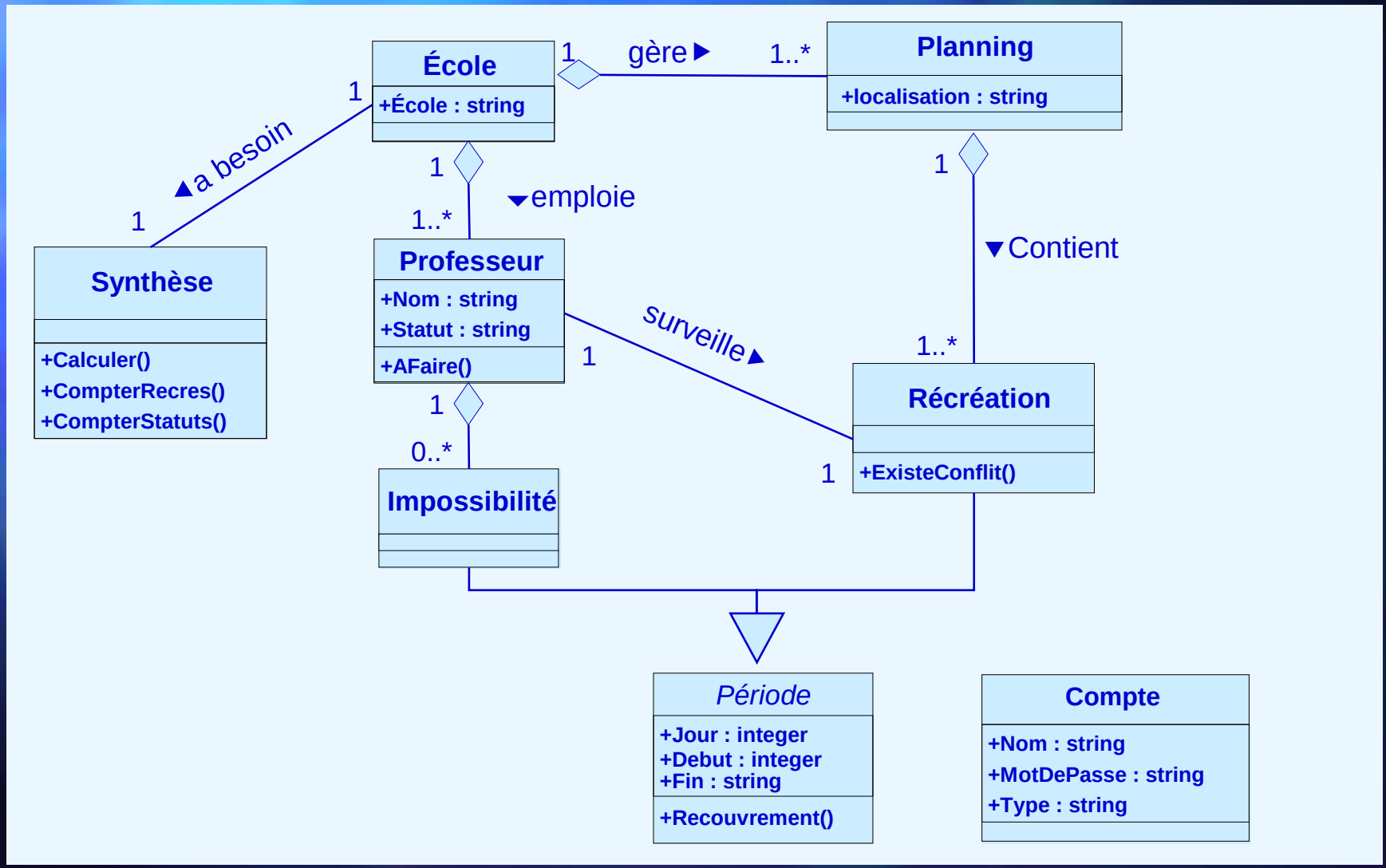


Diagramme de classes d'analyse

Figure 8



Dictionnaire de données de l'analyse

Compte : **Un compte utilisateur.**

- **Attributs :**
  - **Nom : nom de l'utilisateur.**
  - **Mot de passe : mot de passe de l'utilisateur.**
  - **Type : indique si le compte appartient à un planificateur ou à un gestionnaire du personnel.**

Impossibilité : **une période de temps pendant laquelle un professeur ne peut surveiller aucune récréation. Sous-classe de Période.**

Dictionnaire de données de l'analyse

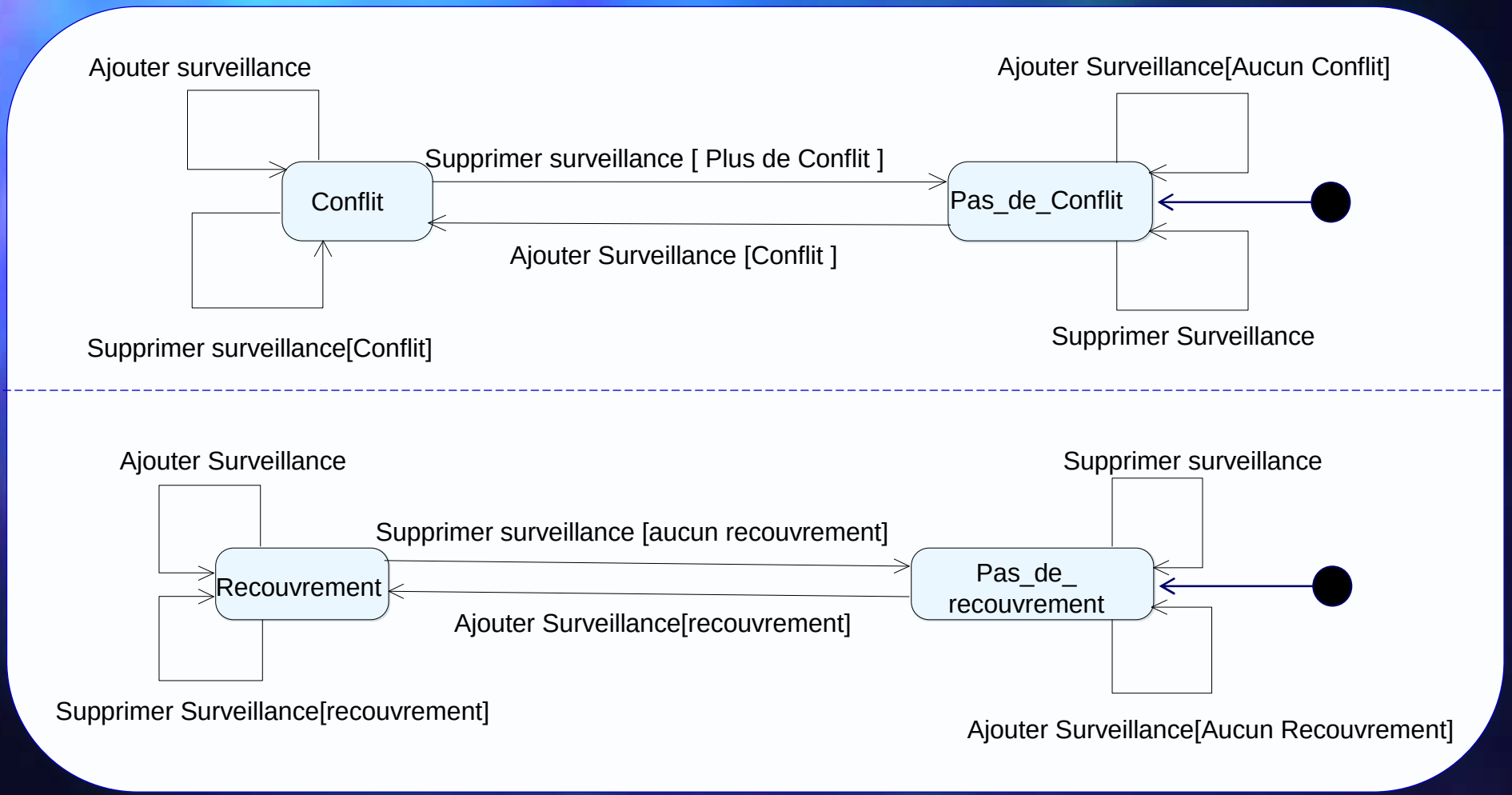
- ***Synthèse :*** **la synthèse est calculée pour l'Organisateur et montre à combien de récréations un professeur est déjà affecté, combien de récréations il (elle) devrait surveiller et combien de récréations il y a à surveiller. Il existe exactement une instance de synthèse.**
  - **Opérations :**
    - **Calculer () : calcule la synthèse.**
    - **CompterRécrés () : compte le nombre total de récréations de tous les plannings de l'Organisateur.**
    - **CompterStatuts () : compte le nombre total de "temps pleins" pour tous les professeurs (voir la classe Professeur).**

Problèmes possibles d'affectations de surveillances

- **Recouvrement :** la récréation affectée au professeur interfère avec l'une de ses périodes d'impossibilité.
- **Trop/trop peu d'affectations :** le nombre de récréations auxquelles est affecté un professeur est trop grand ou trop petit, vis à vis de ses obligations.
- **Conflit :** un professeur est affecté à 2 récréations au même moment.

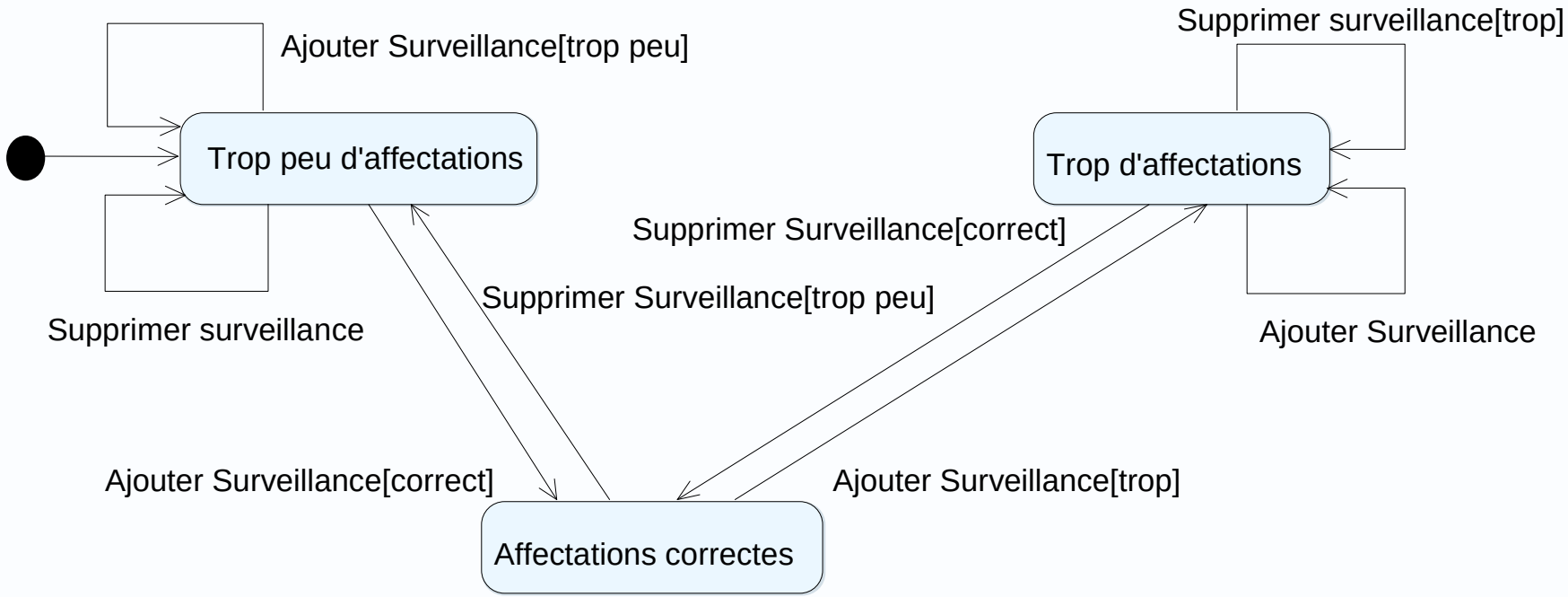
Diagrammes d'états de la classe Professeur

Figure 9



Diagrammes d'états de la classe Professeur

Figure 9



Conception du système

- Certaines classes du domaine sont conservées sans changement de nom, de fonctionnalités ou d'attributs.
- Des classes du domaine peuvent être supprimées (concepts qui ne sont pas implémentés au moyen de classes dans le futur système).
- Des classes peuvent être fusionnées ou éclatées :
  - pour optimiser les accès,
  - pour cacher des données pour des raisons de sécurité
  - pour raffiner des classes d'analyse complexes.
- Des classes, attributs et opérations nécessaires pour modéliser des concepts techniques de l'implémentation choisie peuvent être introduits.

Cas de la classe Synthèse : problème

- Plus une collection de fonctions spécialisées, qu'une véritable classe :
  - elle n'a aucun attribut,
  - ne participe à aucune association non triviale,
  - n'est pas utilisée pour contenir des données.
- cette classe calcule un certain nombre d'attributs d'autres classes.

Cas de la classe Synthèse : solutions

- Mettre en place une classe synthétique responsable des fonctionnalités :
  - Avantage : la représentation de cette fonctionnalité est centralisée et peut être facilement compréhensible par d'autres.
  - Inconvénient : cette classe nécessite de nombreuses références à d'autres classes et devient donc très sensible aux changements.
- Eclater les fonctions parmi les classes auxquelles elles appartiennent logiquement :
  - Avantage : conception simple et directe.
  - Inconvénient : cette fonctionnalité est maintenant dispersée dans plusieurs classes (conception moins claire).

Cas de la classe Synthèse : solution

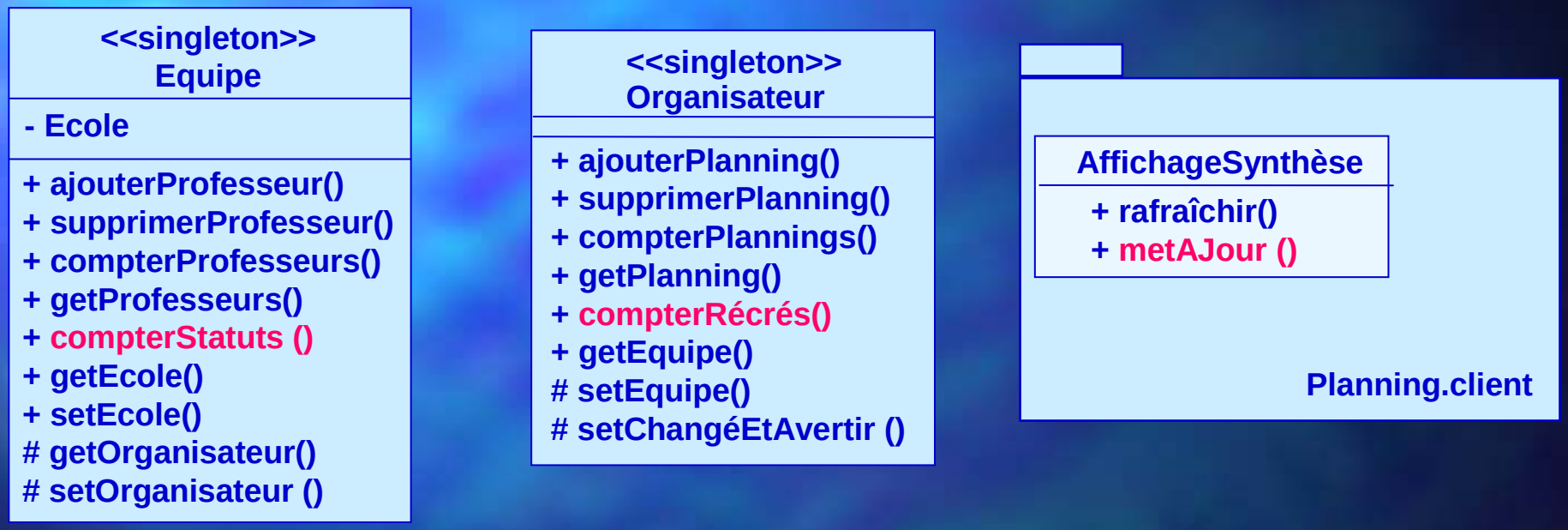


Diagramme de Classes de conception- Figure 10

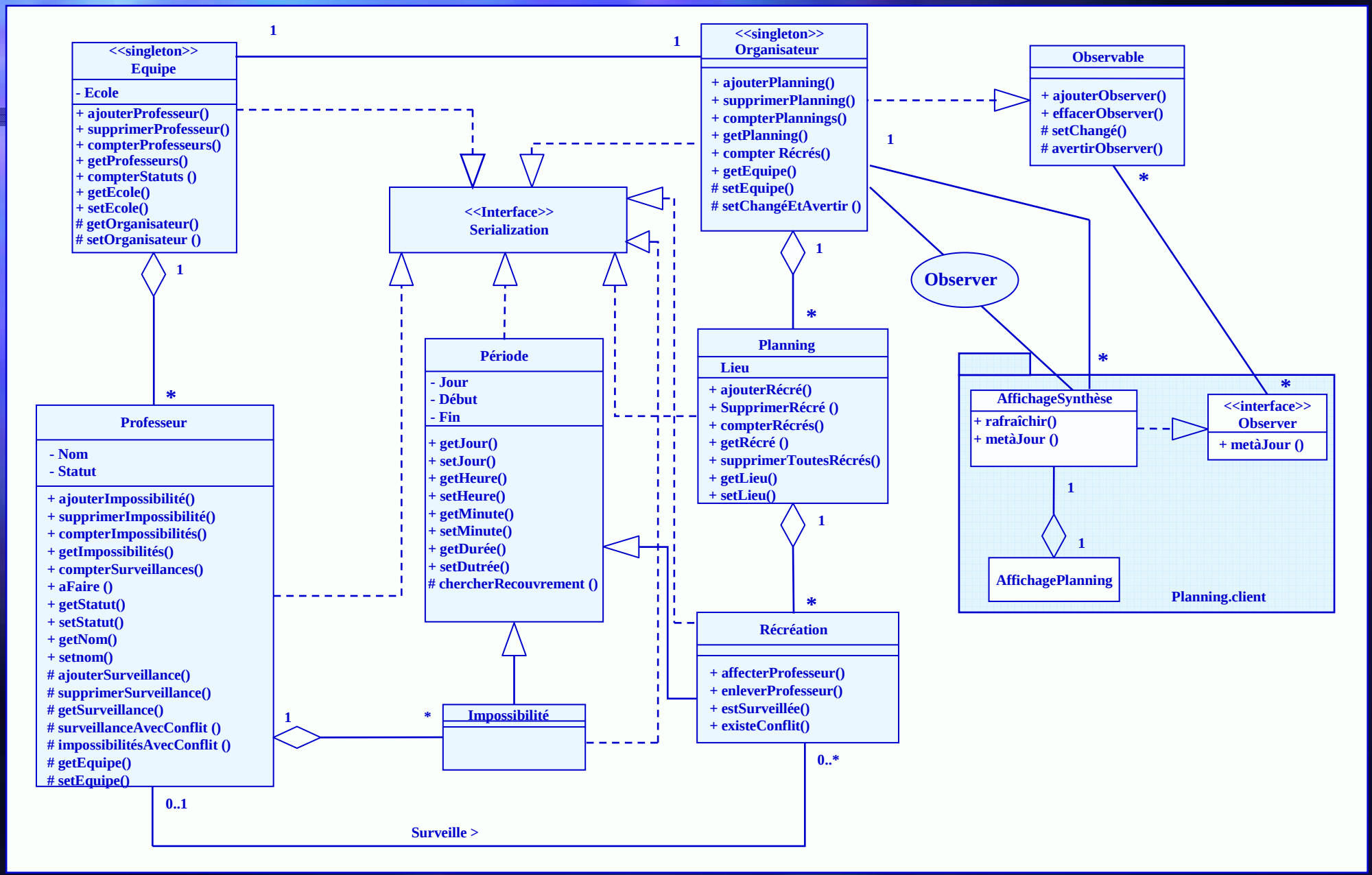


Diagramme de collaboration du patron Observer

Figure 11

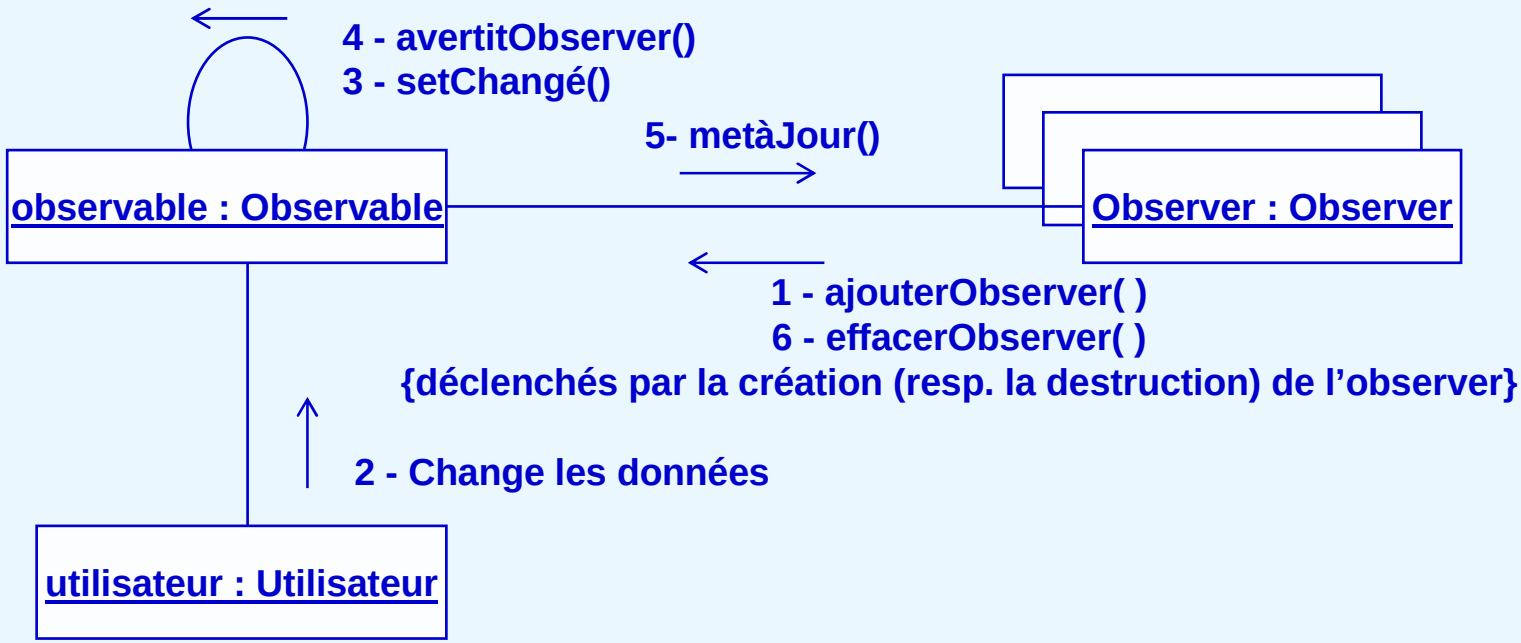
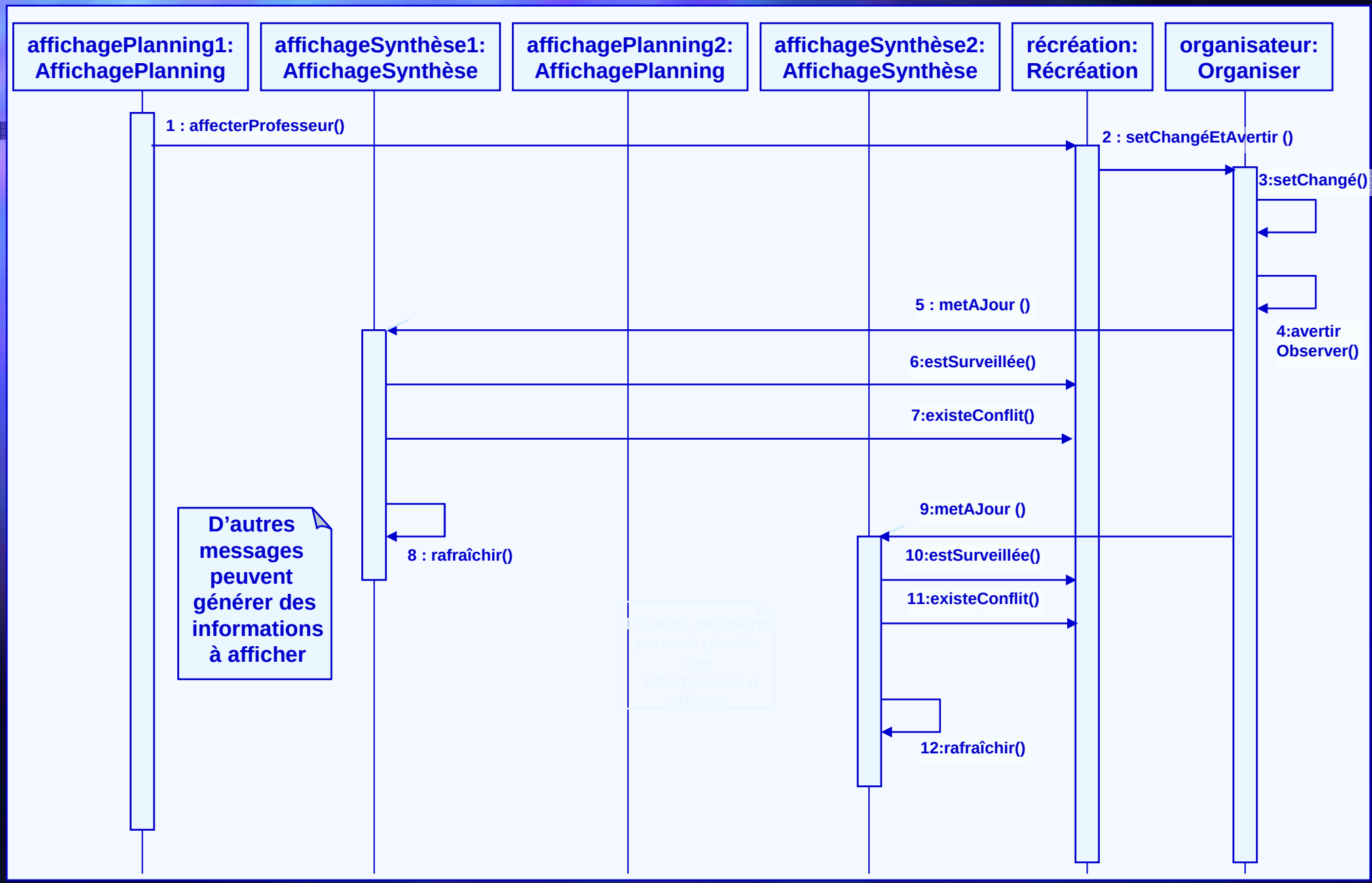


Diagramme de séquence : Mise à jour à chaque changement - Figure 12



Gestion des associations

- La responsabilité de la création et de la destruction d'associations doit être assignée à certaines classes :
  - Souvent l'une des classes associées gère l'association en exclusivité,
  - Il est aussi possible que les deux classes (ou même d'autres) soient impliquées.
- Les directions des chemins d'accès des associations doivent être fixées :
  - Des accès unidirectionnels sont suffisants pour la plupart des associations et l'orientation va de la classe "dirigeante (gérante)" vers la classe associée.

Cas Planning

- la plupart des associations sont des agrégations gérées par l'objet composite :
  - orientées de l'objet composite vers l'objet contenu.
- Les associations entre Equipe et Organisateur, et l'association "Surveille" entre Professeur et Récréation sont toutes deux bidirectionnelles :
  - une Récréation a besoin d'un lien vers le Professeur qui la surveille pour déterminer si cette affectation provoque un conflit avec d'autres récréations du professeur.
  - Un Professeur a besoin d'un lien vers l'Organisateur pour calculer le montant de surveillances qu'il doit assurer.

Dictionnaire de Données du diagramme de classes de conception

- **Interfaces**
  - ***Observer*** **:** un objet implémentant l'interface Observer peut s'enregistrer chez les objets Observable. Les objets Observable avertissent l'objet Observer si nécessaire.
    - ***Opérations Publiques :***

    metAJour () indique que l'état d'un ou plusieurs objets Observable a changé.
  - ***Serialization*** **:** les objets implémentant l'interface Serialization peuvent être stockés dans un ObjectStream. L'envoi du flot dans un fichier rend les objets persistants.
- **Classes**
  - ***AffichageSynthèse :*** une classe de l'interface graphique, représentant la fenêtre de synthèse du client de l'application Planification. **Implémente Observer.**
  - *Opérations publiques* :

  rafraîchir() : ré-affiche la fenêtre de synthèse sur l'écran.

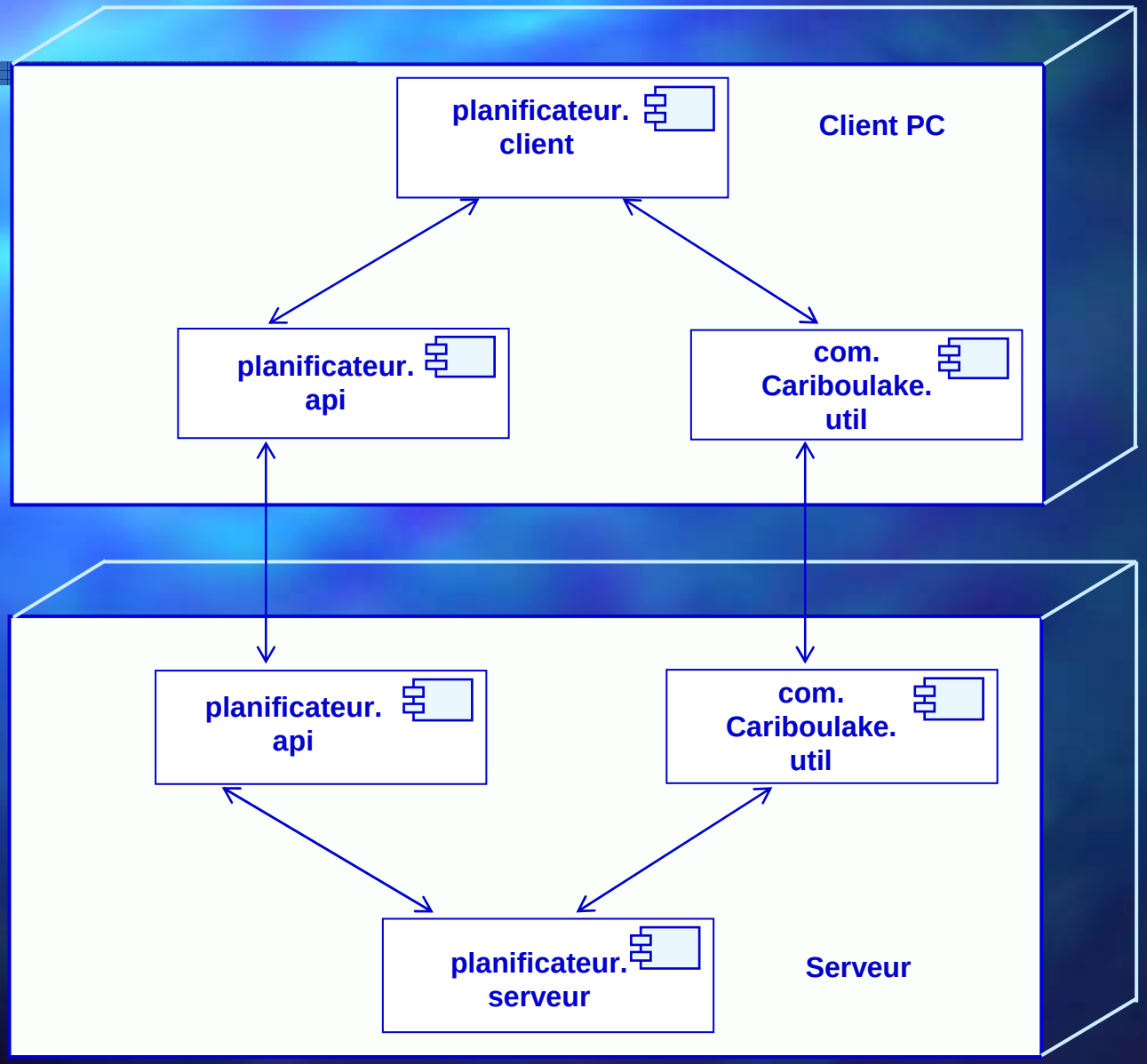
  metAJour () : met à jour la fenêtre synthèse quand les données gérées par l'organisateur ont été changées. Pour cela, la méthode calcule d'abord les nouvelles valeurs pour la synthèse (voir description de la méthode calculer () de la classe Synthèse dans le dictionnaire de données du diagramme de classes du domaine) et ensuite appelle rafraîchir ().

Conception de la distribution

- Répartition des données et des fonctionnalités d'un système sur un réseau (contraintes matérielles).
- *Système cible*
  - Les objets graphiques doivent être gérés par des applets s'exécutant sur des machines clientes.
  - Cette exigence implique aussi l'existence d'au moins un serveur avec une application Java capable de gérer les données persistantes.
- *Partition des objets de l'application*
- Simple solution de type check-in/check-out pour les plannings : les utilisateurs extraient un planning d'un référentiel centralisé, l'édite localement, puis l'enregistrent.
- Chaque utilisateur a besoin que la synthèse soit à jour. Si un utilisateur affecte un professeur à un planning, tous les autres utilisateurs doivent en être informés.
- ⇒Conserver tous les objets de l'application sur le serveur : seules des vues sont chez le client.

Diagramme de composants

Figure 14



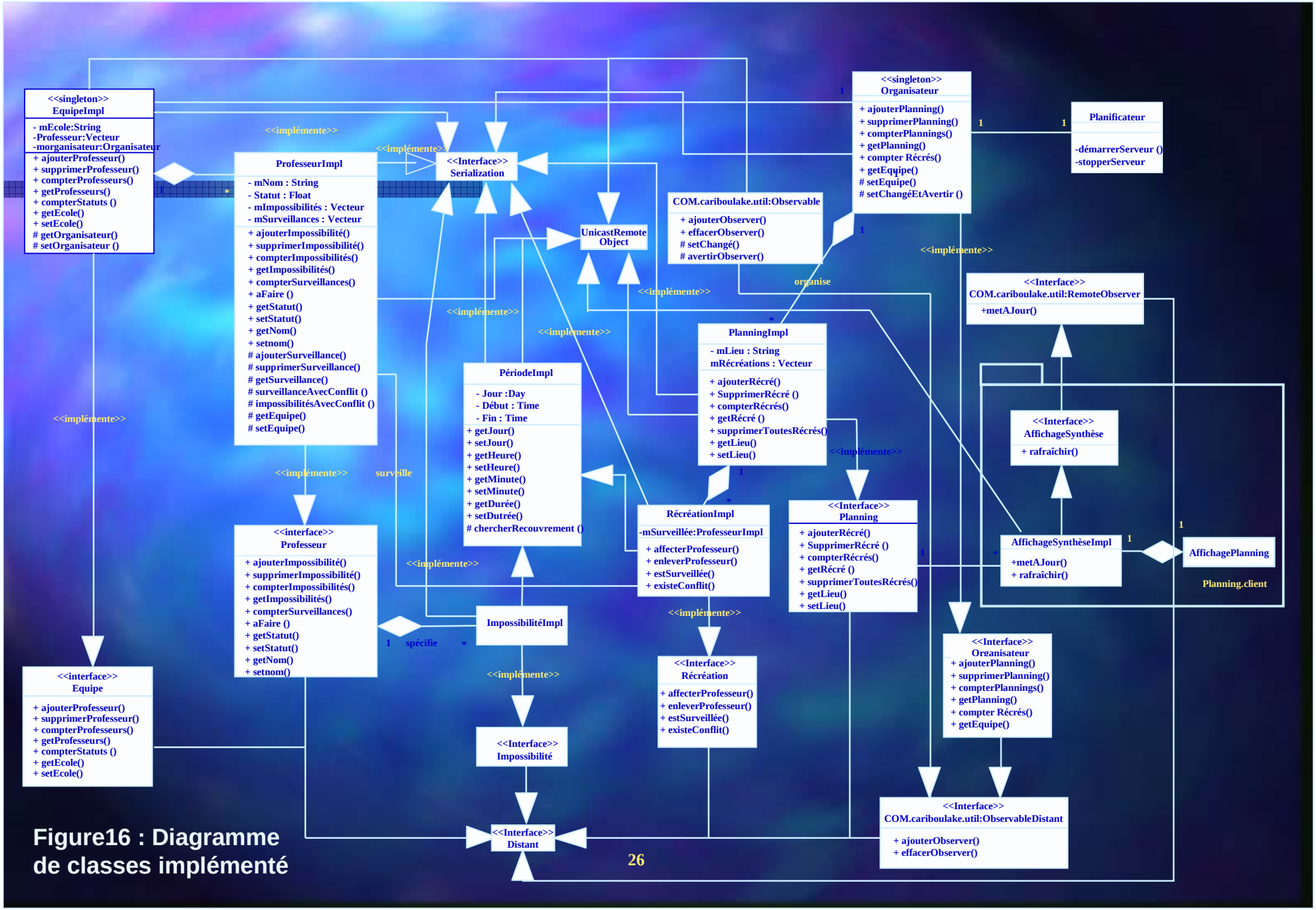


Figure16 : Diagramme de classes implémenté

Synthèse : de l'Analyse à la Conception

- Les classes de conception représentent des classes logicielles et non des des concepts du monde réel,
- Le modèle de conception contient plusieurs types de diagramme (interactions, classes et paquetages),
- Le modèle des cas d'utilisation pilote la conception.

Le modèle de Classes de Conception

- Identifier les classes architecturalement significatives (trace / domaine, Diagrammes d'Interactions)
- Prendre en compte les principes de faible couplage et de forte cohésion,
- Rechercher les Patrons architecturaux, les composants réutilisables,
- Ajouter les classes techniques (maîtresse, interfaces ^{2}, conteneurs…),
- Déterminer les classes persistantes et leur moyen de stockage,
- Regrouper les classes dans des paquetages (sous-systèmes) organisés en couches.
- Détailler attributs, méthodes, exceptions, visibilité, navigabilité.