

4. Diagramme de collaboration

Le diagramme de collaboration fait partie des vues dynamiques du système d'information.

4.1 Origine

Les diagrammes de collaboration proviennent d'une adaptation des diagrammes d'objet de Booch (object diagram), des graphes d'interaction d'objet de Fusion (object interaction graph) et d'autres sources.

4.2 Définition

Ce type de diagramme montre les interactions et les liens entre objets (instances de classes et acteurs). Il s'intéresse à la structure de collaboration entre objets (séquencement, itération, concurrence, etc.). Il permet de représenter le contexte d'une interaction, car on peut y préciser les états des objets qui interagissent.

Les diagrammes de collaboration montrent les interactions entre les objets à travers la représentation chronologique d'envois de messages, mais le temps n'est pas représenté implicitement. La chronologie des interactions est indiquée par la numérotation de messages pour indiquer leur ordre d'envoi.

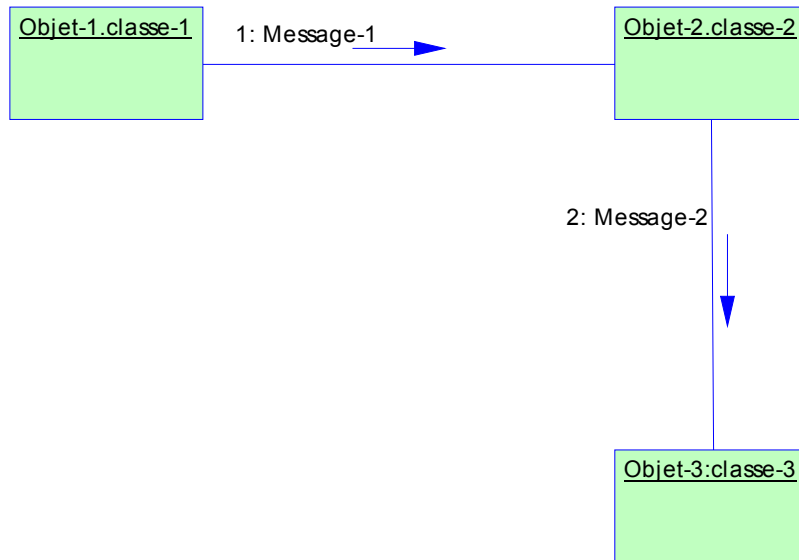
Ce diagramme est équivalent au diagramme de séquences (chapitre 5). Cependant, l'aspect temporel n'apparaît pas, mais l'aspect chronologique est présent.

Le diagramme de collaboration représente ces éléments réalisant un objectif particulier (une fonctionnalité du système). Il peut être utilisé pour illustrer l'exécution d'une opération ou d'un cas d'utilisation, ou simplement un scénario d'interactions dans le système. Il permet de concevoir un exemple d'interactions entre objets.

4.3 Composants du diagramme

Les diagrammes de cas collaboration comportent trois composants :

- Les objets (instances de classe)
- Les liens entre objets (comme dans le diagramme de classe)
- Les interaction entre objets sous forme d'une suite de messages
- Les acteurs



Objet

Un objet est une instance d'une classe. Il peut-être persistant ou transitoire : un objet persistant est un objet qui continue d'exister une fois que le processus qui l'a créé est terminé, un objet transitoire est un objet qui cesse d'exister une fois que le processus qui l'a créé est terminé.

Les objets sont obtenus à partir des classes. Chaque objet est décrit par :

Nom-objet :classificateur ou *nom-objet :nom-classe*.

Un *multiple* définit un jeu d'instances. Il s'agit d'une représentation graphique d'un objet qui présente plusieurs instances, il ne peut en revanche avoir qu'un seul ensemble d'attributs et ce, même s'il représente plusieurs instances. Un objet peut communiquer avec un autre objet qui est un multiple. Cette fonctionnalité est principalement utilisée dans le diagramme de collaboration mais peut également être utilisée dans le diagramme d'objets.

Exemple

Un employé gère une liste de documents : c'est la liste qui représente l'objet multiple.

Liens

Un lien entre objets représente une connexion entre deux objets, il permet de mettre en exergue la collaboration entre objets, d'où le nom de diagramme de collaboration. Il est représenté sous forme d'un trait plein entre :

- · Deux objets
- · Un objet et un acteur (ou vice-versa)

Un lien entre objets peut être une instance d'association entre classes.

Acteurs

La définition d'un acteur dans le contexte d'un diagramme de collaboration est la même que celle en vigueur dans les diagrammes de cas d'utilisation ou de séquence ; elle spécifie un utilisateur externe, ou un ensemble d'utilisateurs liés qui interagissent avec un système.

Message, Synchronisation des messages

Dans un diagramme de collaboration il est possible de spécifier de manière très précise l'ordre et les conditions d'envoi des messages. Pour chaque message, il est possible d'indiquer :

- *son numéro d'ordre*
qui indique le rang du message, c'est-à-dire son numéro d'ordre par rapport aux autres messages. Les messages sont numérotés à l'aide de chiffres séparés par des points. Ainsi, il est possible de représenter le niveau d'emboîtement des messages et leur précedence.
Exemple : l'envoi du message 1.3.5 suit immédiatement celui du message 1.3.4 et ces deux messages font partie du flot (de la famille de messages) 1.3.
Pour représenter l'envoi simultané de deux messages, il suffit de les indexer par une lettre.
Exemple : l'envoi des messages 1.3.a et 1.3.b est simultané.
- *le prédicat qui conditionne son envoi*, spécifiée sous forme d'expression booléenne
- *la liste des messages prédécesseurs*,
composée d'une liste de numéros d'ordre suivie de '/', la liste des prédécesseurs définit quels messages doivent être échangés avant que le message courant ne puisse être envoyé.
Exemple : numéros de séquence 1, 2, 4 avant 3 = '1,2,4/ 3'
- *Heure de début / Heure de fin*,
heures définies par l'utilisateur et utilisées pour définir des contraintes.
Exemple : [heure = midi] 1 : manger()
Ce message n'est envoyé que s'il est midi
- *réccurrence du message*.
Permet de spécifier l'envoi répétitif de messages en mode séquentiel (ou en parallèle, avec " | | ")
Exemple n°1 :
1.3.6 * : ouvrir()
Ce message est envoyé de manière séquentielle un certain nombre de fois.
Exemple n°2 :
3 / *||[i := 1..5] : fermer()
Représente l'envoi en parallèle de 5 messages. Ces messages ne seront envoyés qu'après l'envoi du message 3

4.4 Exemples

Recrutement via un cabinet de recrutement :

Diagramme de classes

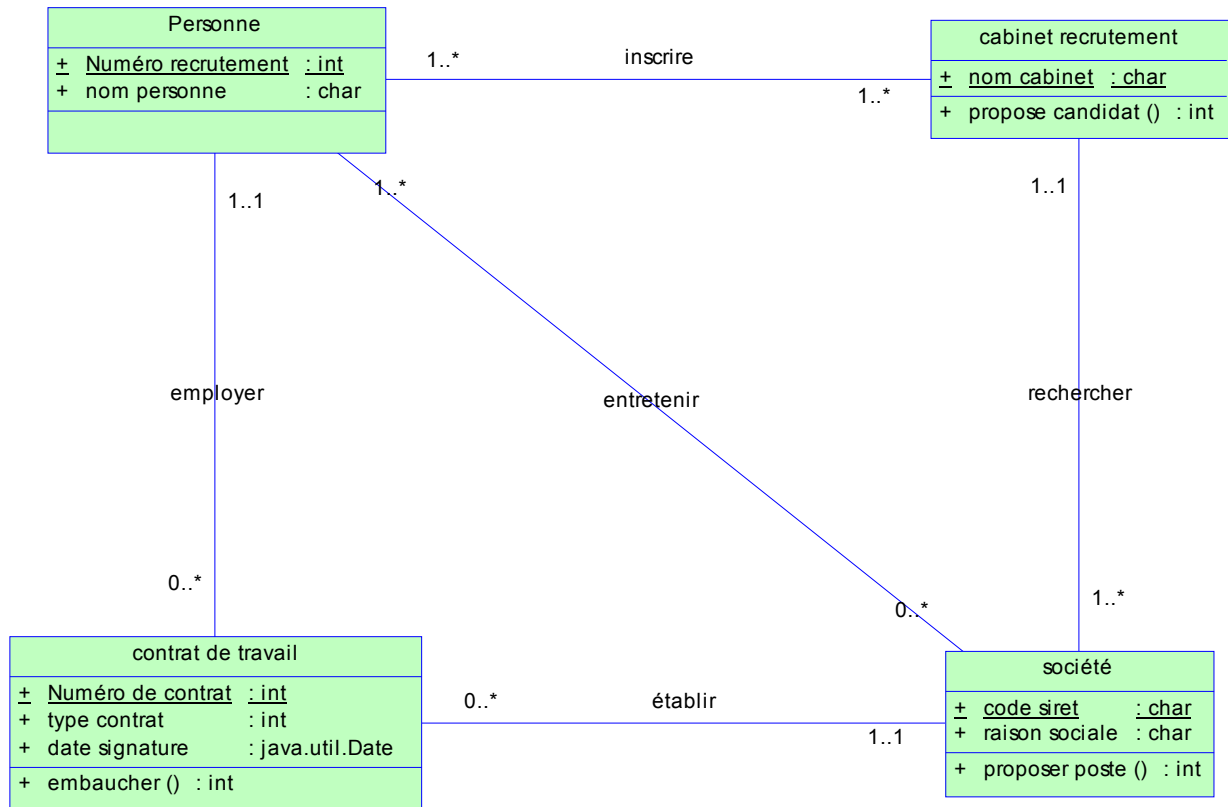


Diagramme de collaboration

