

Intitulé du Master : MASTER PROFESSIONNEL en STIC

1ere année

Modélisation des applications Web

Responsable: Dr. M. BOUZAHZAH

Les phases de modélisation d'une application web

Partie 03

La conception

Introduction

- **La conception commence par le modèle d'analyse et l'architecture en tant qu'apports principaux.**
- **L'activité principale de la conception est d'affiner le modèle d'analyse de sorte qu'il puisse être implémenté avec les composants de l'architecture.**
- **Même si cela semble simple, cela peut être la phase la plus complexe d'un projet de développement, en particulier lorsque des progrès significatifs dans la technologie logicielle se produisent si fréquemment.**

L'étape de conception

- **Les activités de conception tournent autour des diagrammes de classe et d'interaction.**
- **Les classes deviennent plus définies, avec des propriétés pleinement qualifiées (nom et type) et des opérations (signatures complètes).**
- **Des classes supplémentaires, principalement des classes d'assistance et d'implémentation, sont souvent ajoutées lors de la conception.**
- **À la fin, le modèle de conception résultant est quelque chose qui peut être mappé directement dans le code. C'est le lien entre les abstractions de l'entreprise et les réalités du logiciel.**

Les diagrammes de la conception

- Au cours de l'analyse, nous nous sommes contentés de travailler uniquement avec des diagrammes de classes et des diagrammes d'interaction.
- Lors de la conception, un nouveau diagramme - en fait, une nouvelle vue du modèle: la vue des composants - est introduit.

Le diagramme de composant (1)

- Le diagramme de composant exprime le physique - si quelque chose dans le logiciel peut être considéré comme un «physique» - modules exécutables qui seront distribués en tant que système.
- Un composant est généralement mappé vers des fichiers exécutables, des fichiers de classe Java, des bibliothèques statiques ou des bibliothèques de liens dynamiques (DLL).

Le diagramme de composant (2)

- **Un composant est quelque chose qui réalise un ensemble d'interfaces.**
- **Une interface définit le nom de la fonction, ses paramètres et leurs types de données, qu'ils soient facultatifs, qu'ils soient en entrée ou en sortie, et le type de valeur de retour de la fonction.**
- **Un composant peut réaliser plusieurs interfaces.**

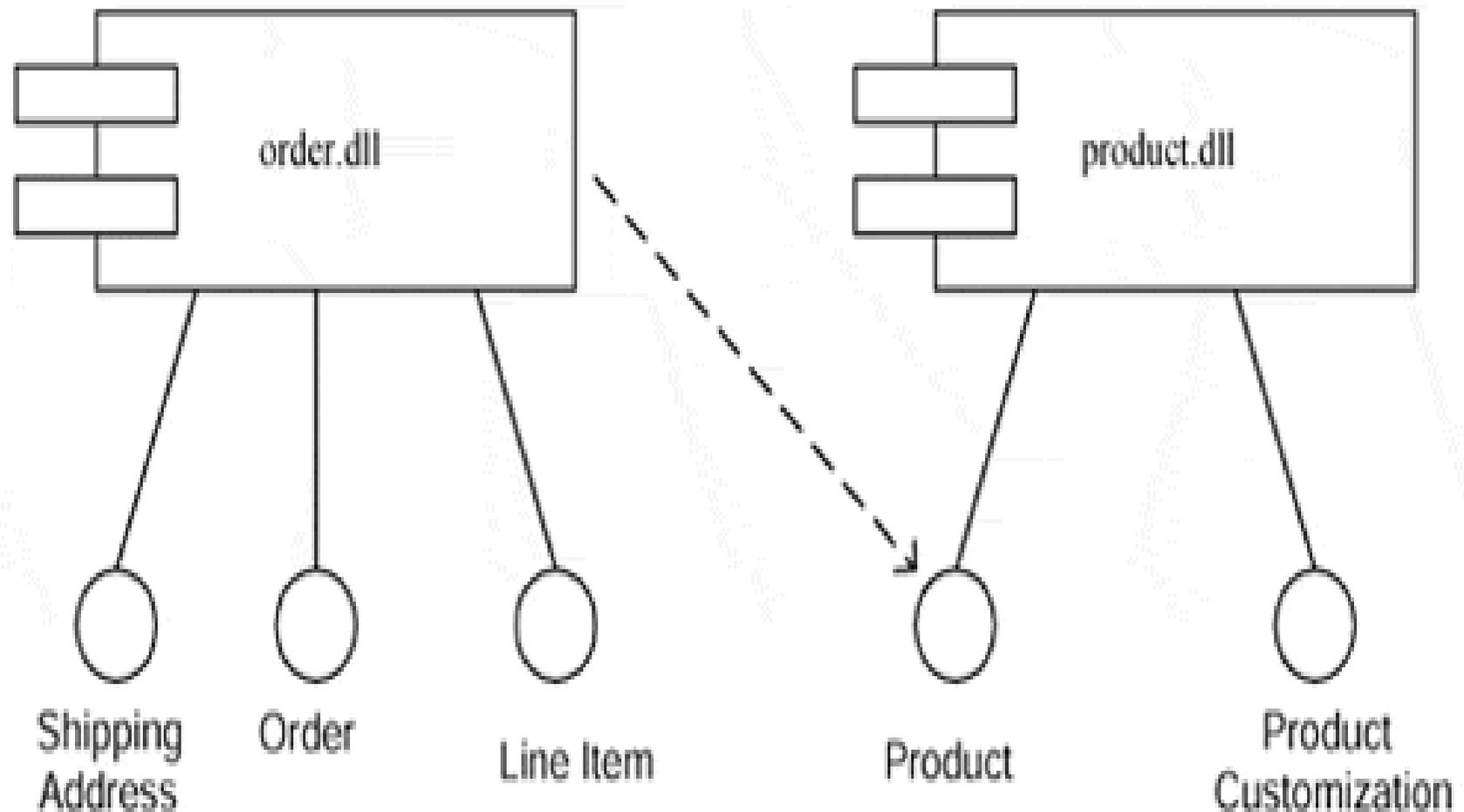
Le diagramme de composant (3)

- La réalisation d'un composant se fait avec les classes et collaborations exprimées dans la vue logique.
- Chaque classe de la vue logique est implémentée par au moins un composant
- Des classes abstraites définissant des interfaces peuvent être implémentées par de nombreux composants.
- Les diagrammes de composants visualisent les composants, les interfaces et leurs interrelations.

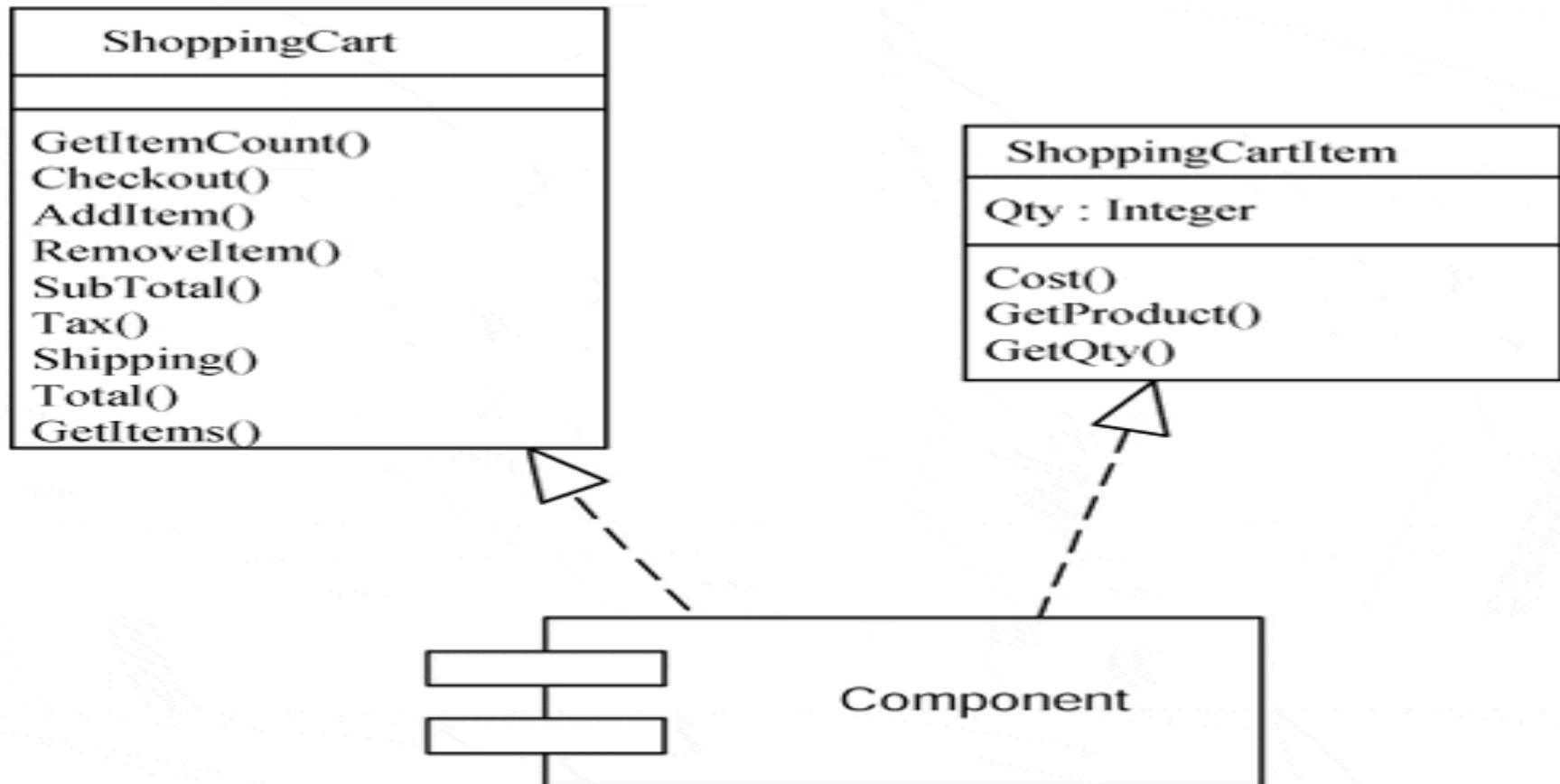
Le diagramme de composant (4)

- **Les composants sont rendus dans un diagramme avec un ensemble de rectangles.**
- **Les interfaces sont rendues avec une "sucette" ou un cercle sur un bâton.**
- **Les dépendances sont indiquées avec des lignes pointillées et des pointes de flèches.**

Le diagramme de composant (5)



Panier d'achat réalise les classes ShoppingCart et ShoppingCartItem.



Les activités de conception

- Lors de la conception, l'interface doit être élaborée en un ensemble d'interfaces spécifiques capables de gérer la communication entre les acteurs et le système.
- les activités de conception comprennent aussi:
 1. Partitionner des objets en niveaux, tels que client, serveur, etc.
 2. Séparer et définir des interfaces utilisateur ou des pages Web (problème de modélisation des pages web scriptées avec UML)

Extension UML pour les applications Web

- **Une extension à UML est exprimée en termes de stéréotypes, de valeurs marquées et de contraintes. Combinés, ces mécanismes nous permettent de créer de nouveaux types de blocs de construction que nous pouvons utiliser dans le modèle.**

Les stéréotypes

- **Un stéréotype représente une extension du vocabulaire de la langue.**
- **Un stéréotype nous permet d'associer une nouvelle signification sémantique à un élément de modèle.**
- **Les stéréotypes sont représentés sous la forme d'une chaîne entre deux guillemets «»**
Cependant, ils peuvent également être rendus par une nouvelle icône.

Les valeurs étiquetées

- La valeur étiquetée est une extension à une propriété d'un élément de modèle.
- Une valeur étiquetée est la définition d'une nouvelle propriété pouvant être associée à un élément de modèle.
- Une valeur étiquetée est rendue sur un diagramme sous la forme d'une chaîne entre crochets.

Les contraintes

- Une contrainte est une extension de la sémantique de la langue.
- Une contrainte est une règle qui définit comment le modèle peut être assemblé.
- Une contrainte spécifie les conditions dans lesquelles le modèle peut être considéré comme «bien formé».
- Les contraintes sont affichées en tant que chaînes entre deux accolades {}.

La conception d'une application web

(1)

- Le partitionnement correct des objets métier dans une application Web est essentiel et dépend beaucoup de l'architecture. Les objets peuvent résider exclusivement sur le serveur, le client ou les deux.
- Certains objets, tels que Facture, peuvent avoir des vies dans les deux. Un objet de facture client peut exister sur le serveur. Cet objet pourrait être envoyé sous la forme d'un document XML au client. Le document XML pourrait être utilisé comme état de la facture.

La conception d'une application web

(2)

- Lorsque le travail est à portée de main, le partitionnement des objets est simple.
- Les applications basées sur l'architecture client web léger placent tous les objets derrière le serveur
- Les applications client web dynamique (lourd) permettent à certains objets de s'exécuter sur le client
- Les applications de livraison Web ont le plus de liberté dans le placement des objets, car ce sont essentiellement des systèmes d'objets distribués qui utilisent simplement un navigateur.

Partitionnement d'objets pour les applications Web client web dynamique(1)

- **Lors du premier passage, les objets persistants, les objets conteneur, les objets partagés et les objets complexes appartiennent tous au serveur.**
- **Les objets associés à des ressources de serveur telles que des bases de données et des systèmes hérités appartiennent également au niveau serveur.**
- **Les objets qui gèrent des associations statiques ou des dépendances avec l'un de ces objets doivent également exister sur le serveur.**

Partitionnement d'objets pour les applications Web client web dynamique(2)

- **Il est plus facile d'identifier les objets pouvant exister sur le client.**
- **Si un objet n'a pas d'associations ou de dépendances avec des objets sur le serveur et possède des associations et des dépendances uniquement avec d'autres ressources client, telles que les navigateurs et les applets Java, il peut exister sur le client.**
- **Les objets candidats pour le partitionnement sur le client sont des objets de validation de champ, des contrôles d'interface utilisateur et des contrôles d'assistance à la navigation.**

Partitionnement d'objets pour les applications objets distribués(1)

- **Le modèle d'architecture livraison web(web delivery) est essentiellement un système d'objets distribués basé sur un site Web.**
- **Ce type d'application utilise des protocoles de communication client et serveur autres que HTTP.**
- **Les objets réels peuvent s'exécuter dans le contexte du client ou du navigateur et ont donc accès à ses ressources.**

Partitionnement d'objets pour les applications objets distribués(2)

- **La partition d'objets pour ce type d'architecture dépend principalement de la nature des objets individuels.**
- **L'une des raisons principales de la distribution d'objets au client est de retirer une partie de la charge du serveur.**
- **Il est également naturel de placer les objets là où ils seront le plus efficaces dans le système.**

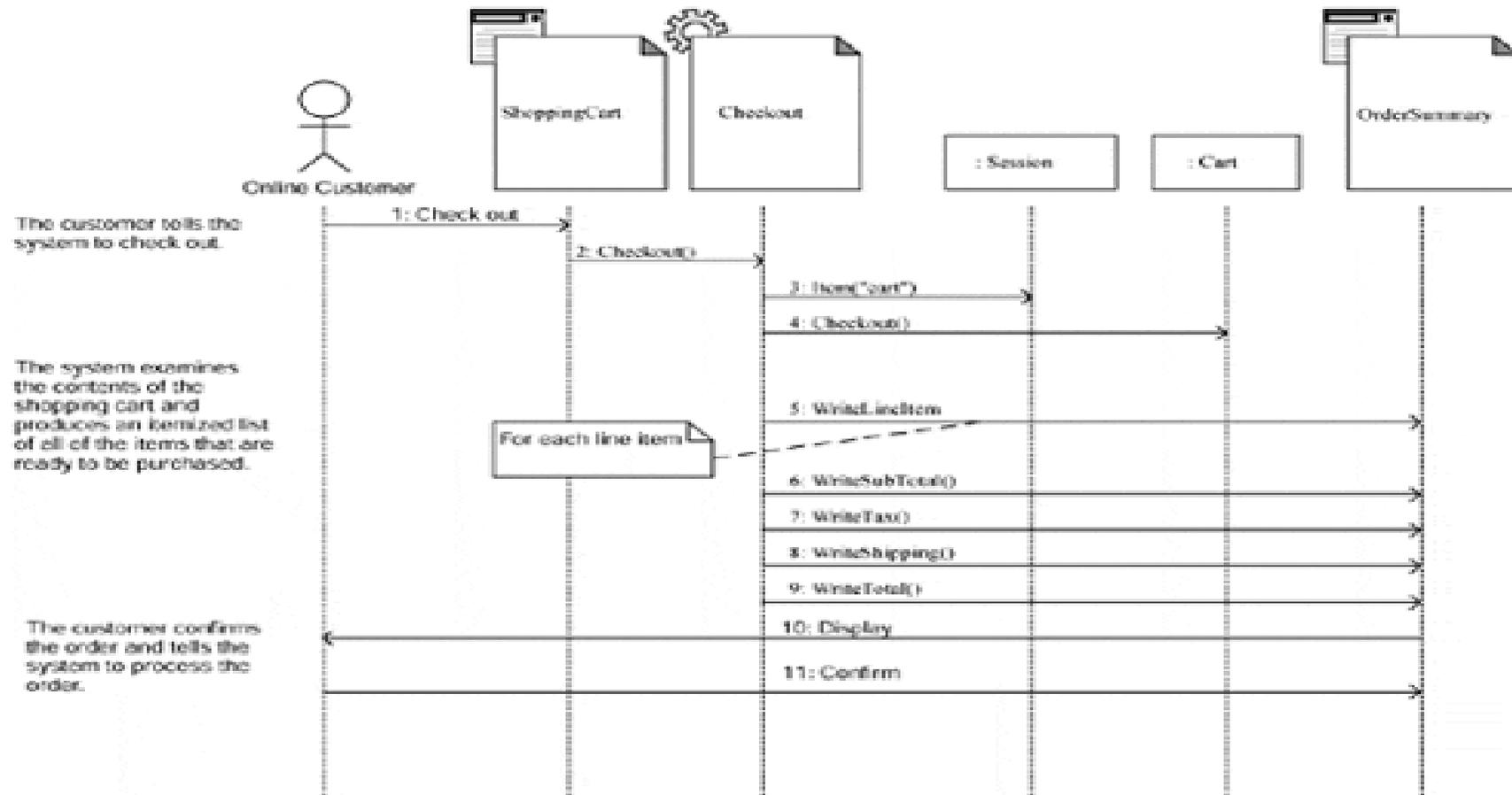
Les diagrammes de séquences et la conception

- **La définition des pages web implique la découverte de pages Web et de leurs relations entre elles et avec les objets du système.**
- **Cette étape dépend aussi fortement du modèle architectural de l'application**
- **Dans cette activité, les objets Système des diagrammes de séquence créés lors de l'analyse évoluent en objets et en pages Web, l'interface utilisateur principale de l'application Web.**

La conception avec le modèle client web léger

- Dans les applications client web léger, les acteurs n'interagissent qu'avec les pages client et les pages du serveur n'interagissent qu'avec les ressources du serveur.
- Par conséquent, il faut mettre une page client et serveur dans le diagramme de séquence.
- La solution la plus simple consiste à transformer directement les objets frontières du modèle d'analyse en pages client et à transformer les objets du contrôleur en pages serveur.

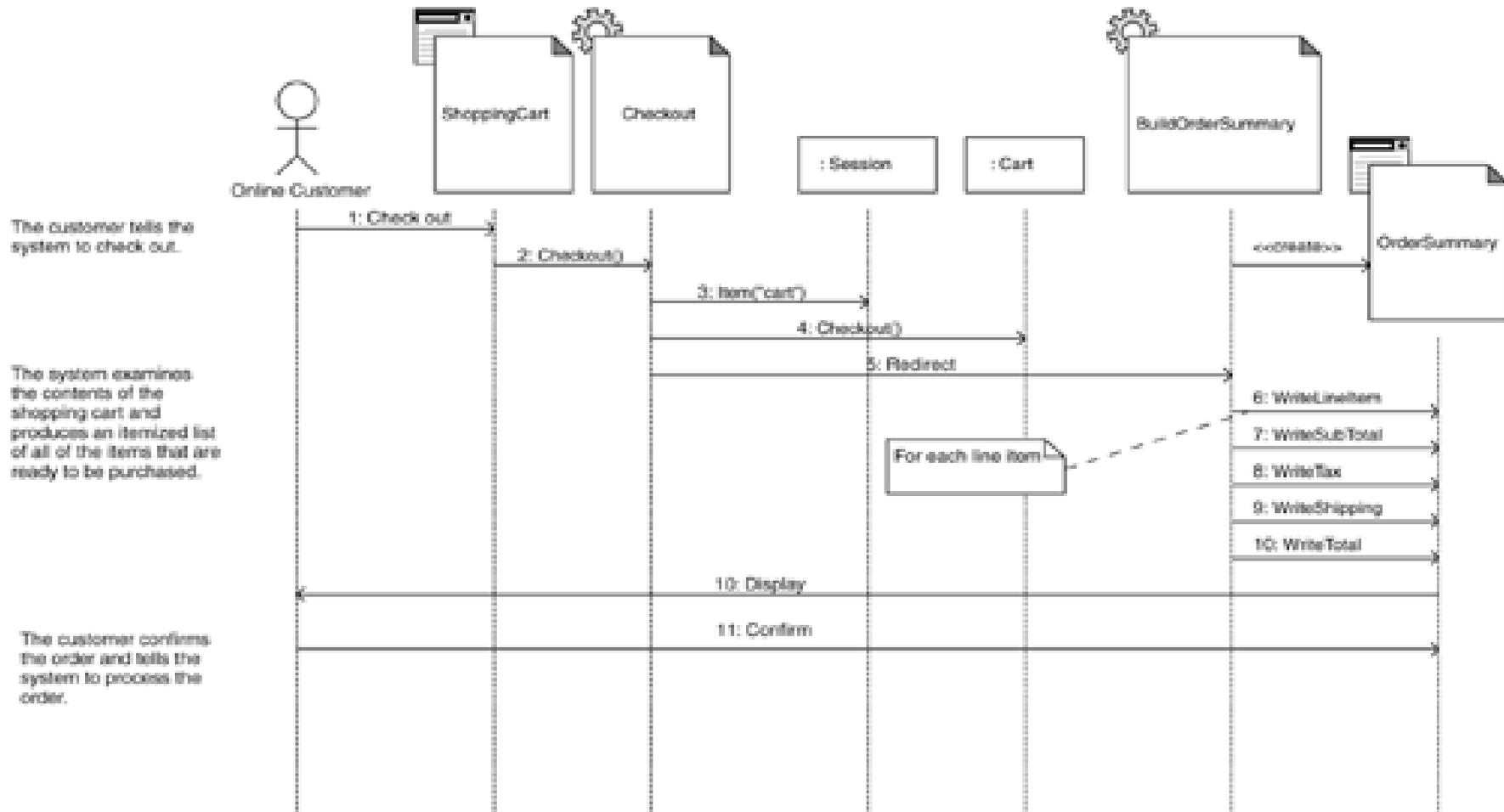
Le diagramme de séquence élaboré pour l'opération de paiement



Le diagramme de séquence

- Une grande partie de l'activité détaillée du processus de vérification n'est pas représentée dans le diagramme
- mais peut être exprimée dans des diagrammes supplémentaires. Ce type d'activité peut inclure la création de transactions, ou d'objets intermédiaires, nécessaires pour vérifier un panier.

Séparer les responsabilités de la page du serveur



Explication

- Les parties restantes de ce scénario simple commencent par la nouvelle page client, OrderSummary, car elle a remplacé la page Commander dans le navigateur du client.
- Dans le scénario, l'acteur confirme le processus de commande en envoyant un message à la page OrderSummary, puisque c'est la page maintenant disponible pour l'acteur. Ce message est aussi probablement un lien hypertexte vers une autre page du serveur.

La vue logique des classes liées à la vérification d'un panier

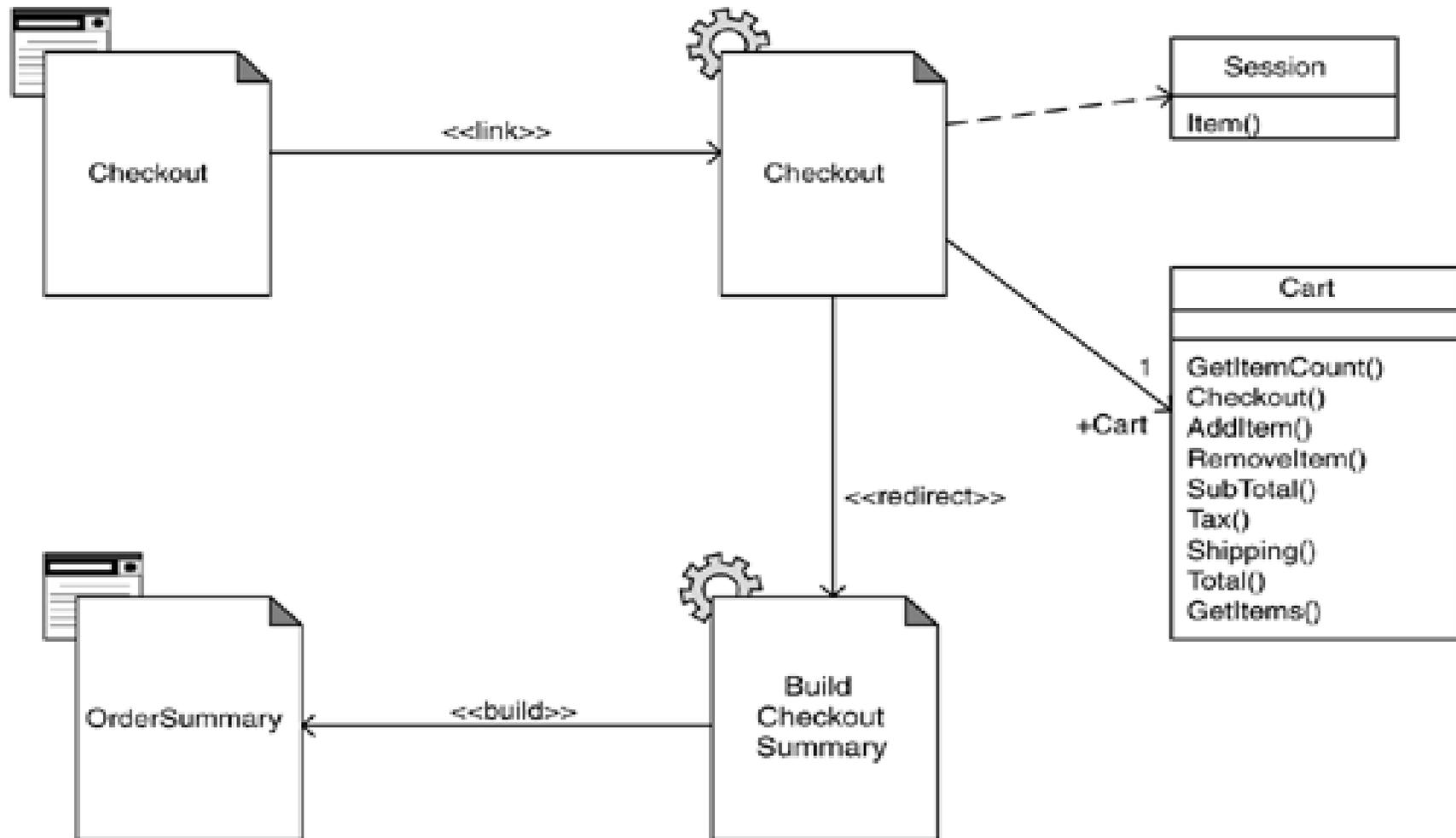
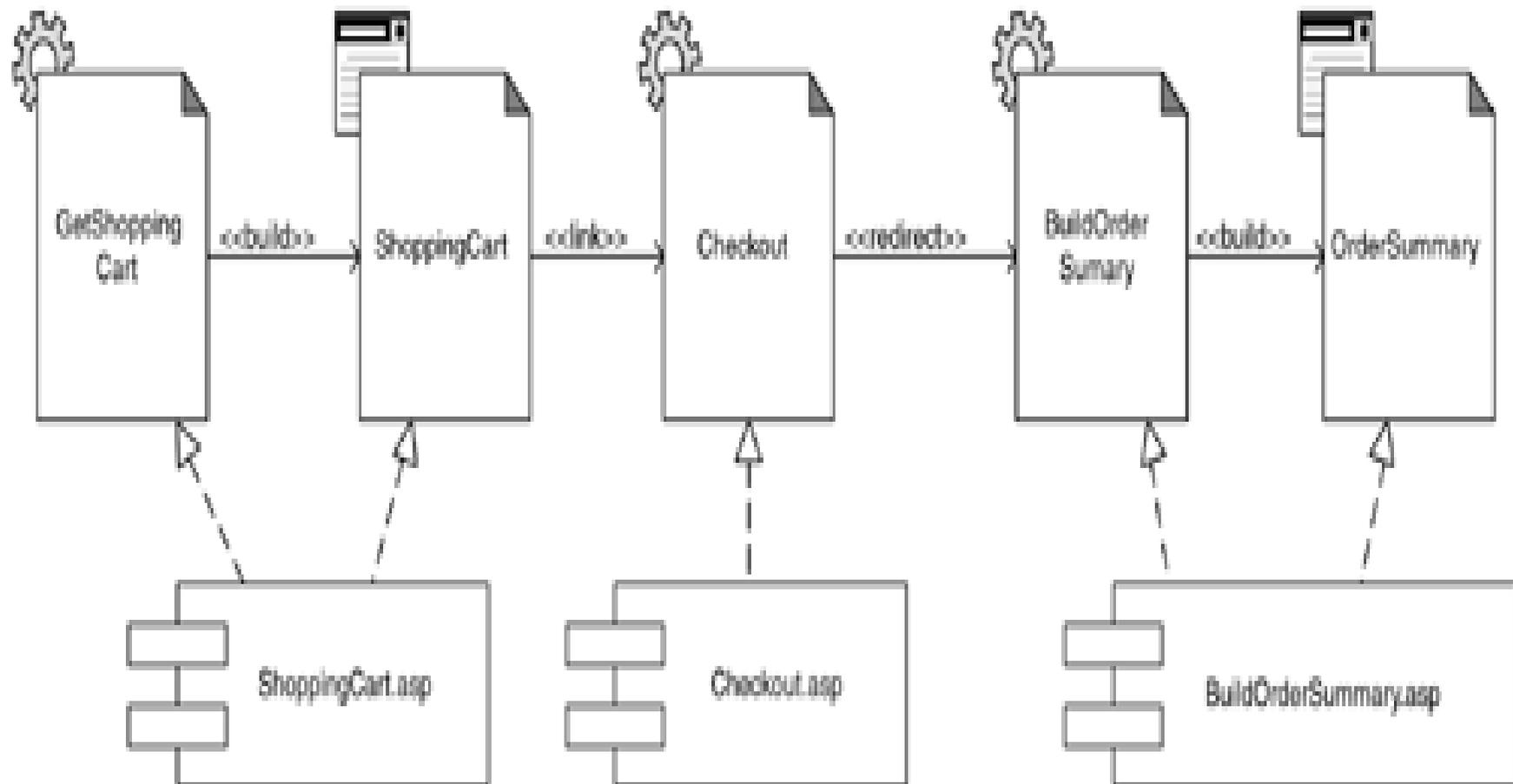


Diagramme de composants

- En fin de compte, ces pages Web conceptuelles (serveur et client) doivent se retrouver dans un composant du système.
- Le diagramme qui suit montre trois composants, chacun représentant une page Web (ShoppingCart.asp, Checkout.asp et BuildOrderSummary.asp) qui sera fournie avec le système final.
- Ce diagramme est une réalisations de composants des pages de paiement

Réalisations de composants des pages de paiement



La conception avec le modèle client web dynamique (1)

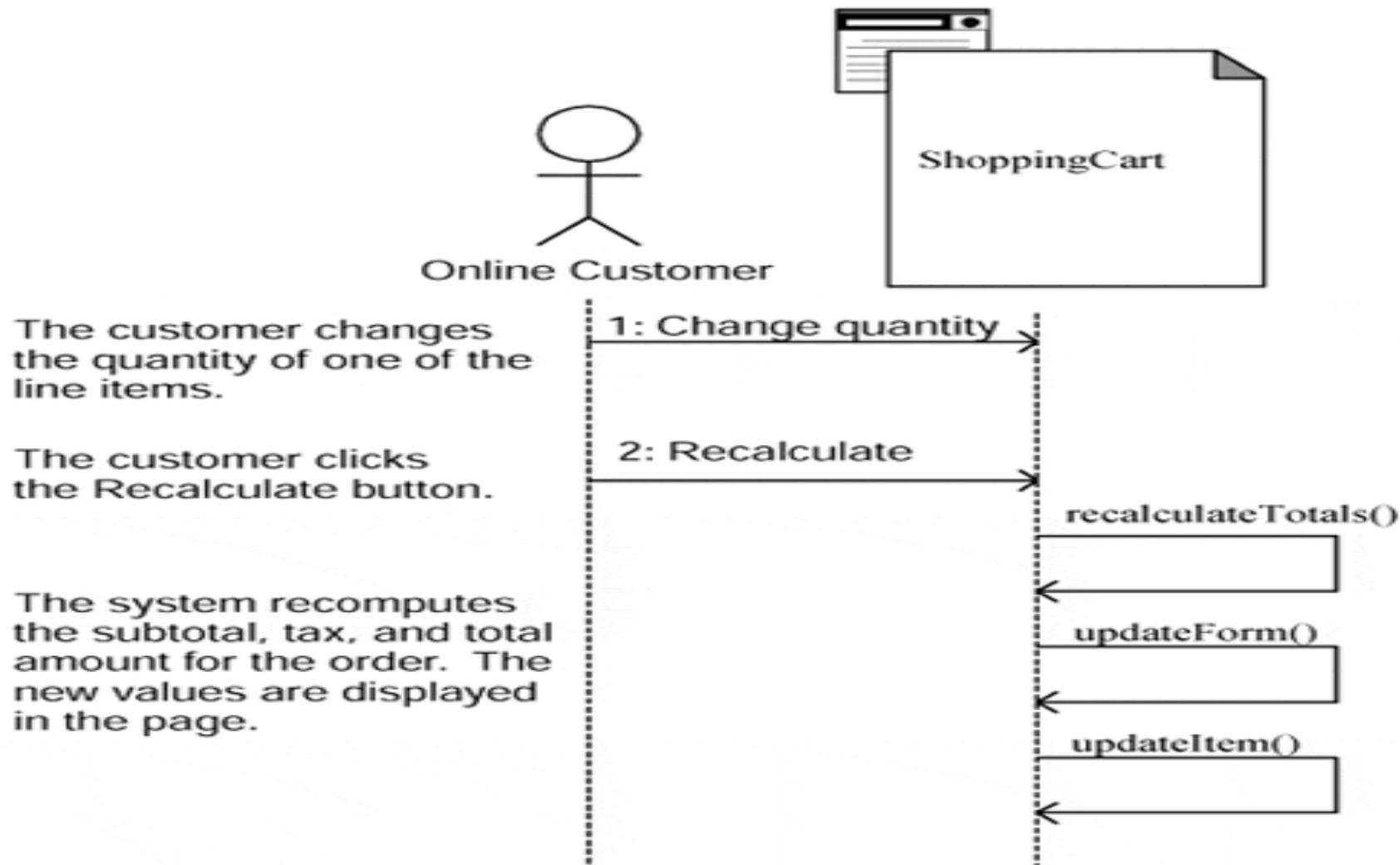
- **La conception d'applications Web comportant des pages client dynamiques nécessite une attention particulière au partitionnement des objets.**
- **Dans ce type d'applications, il n'y avait aucune tentation de dessiner des associations entre les objets client et les objets serveur, car il n'y a pratiquement aucun objet client définissable par l'utilisateur**
- **Ce type d'applications peuvent cependant avoir toutes sortes d'objets et d'activités sur le client.**

La conception avec le modèle client web dynamique (2)

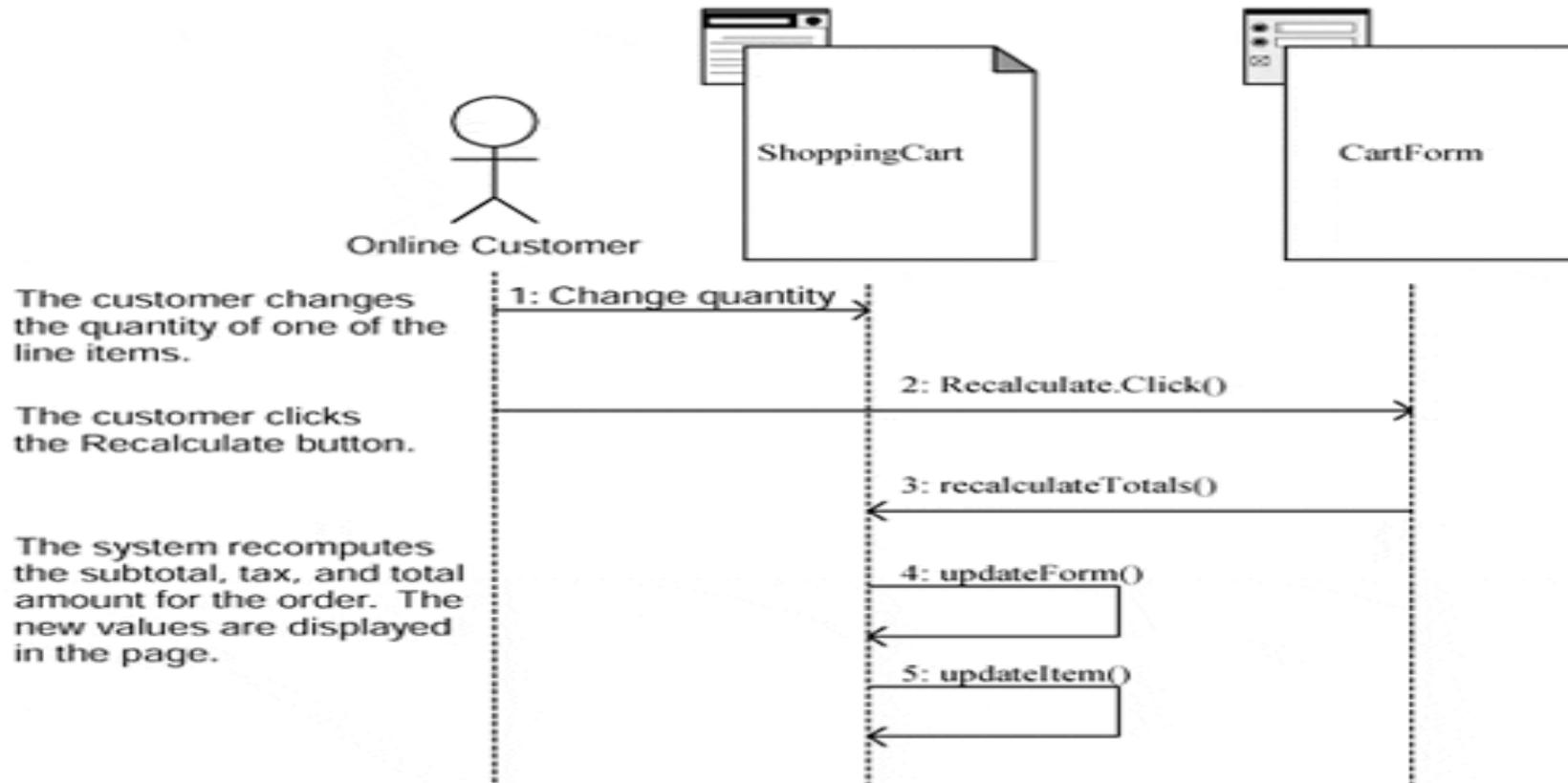
1. Les scripts coté client

- La conception des systèmes client web dynamique commence par les diagrammes de séquence produits avec les cas d'utilisation.
- Une page client peut exécuter des opérations sur elle-même
- Les opérations sont des fonctions JavaScript dans la page

Exprimer des scripts dans une page client



Manière alternative pour exprimer un script dans une page client



La conception avec le modèle client web dynamique (2)

2. Objets clients

- La fonctionnalité peut être fournie non seulement par les scripts, mais également avec des objets tels que les contrôles ActiveX, les applets Java et les JavaBeans.
- Ces types de composants, souvent utilisés lorsque des fonctionnalités vraiment sophistiquées sur le client sont requises

Diagramme de classes avec une page client utilisant des composants ActiveX

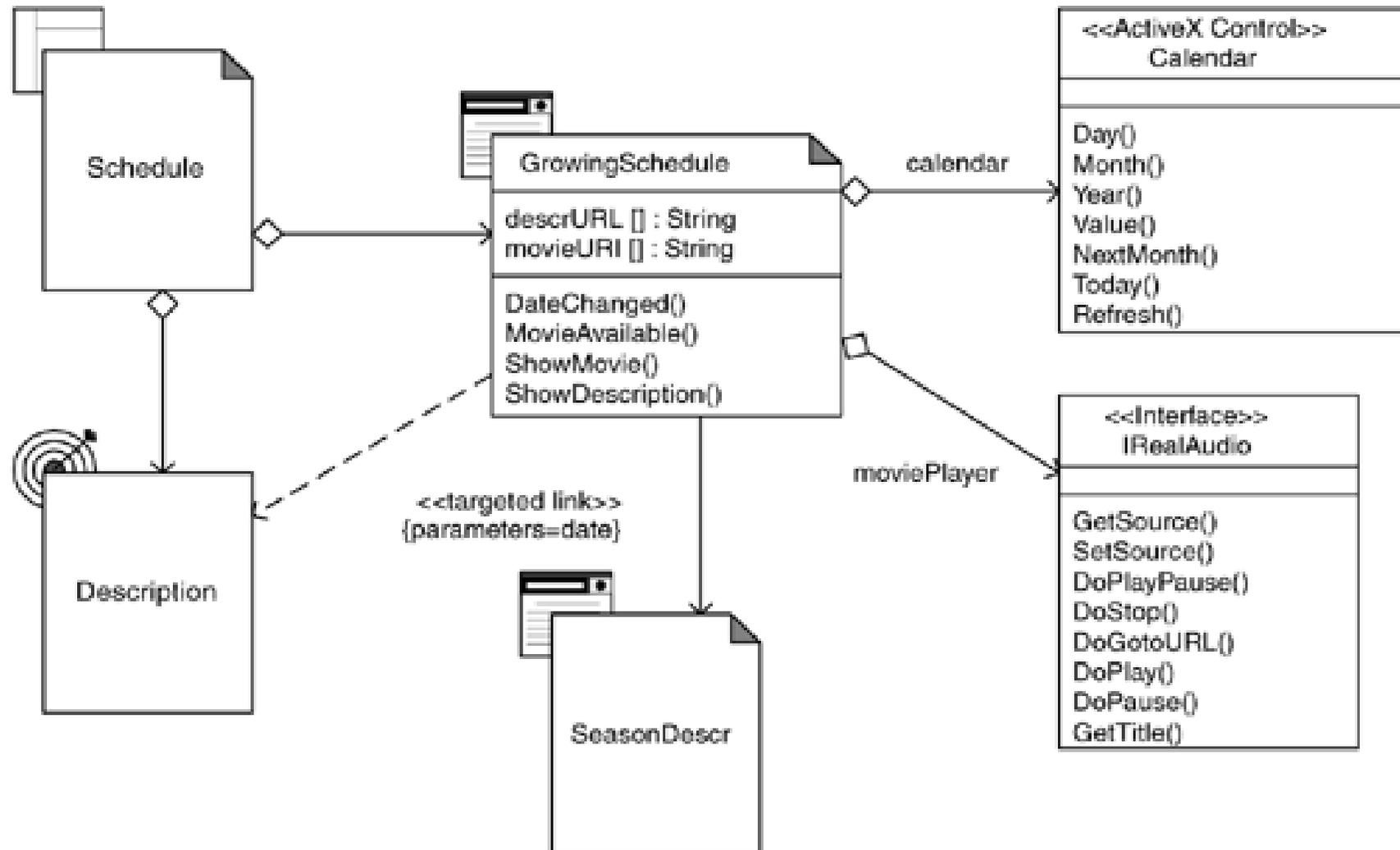
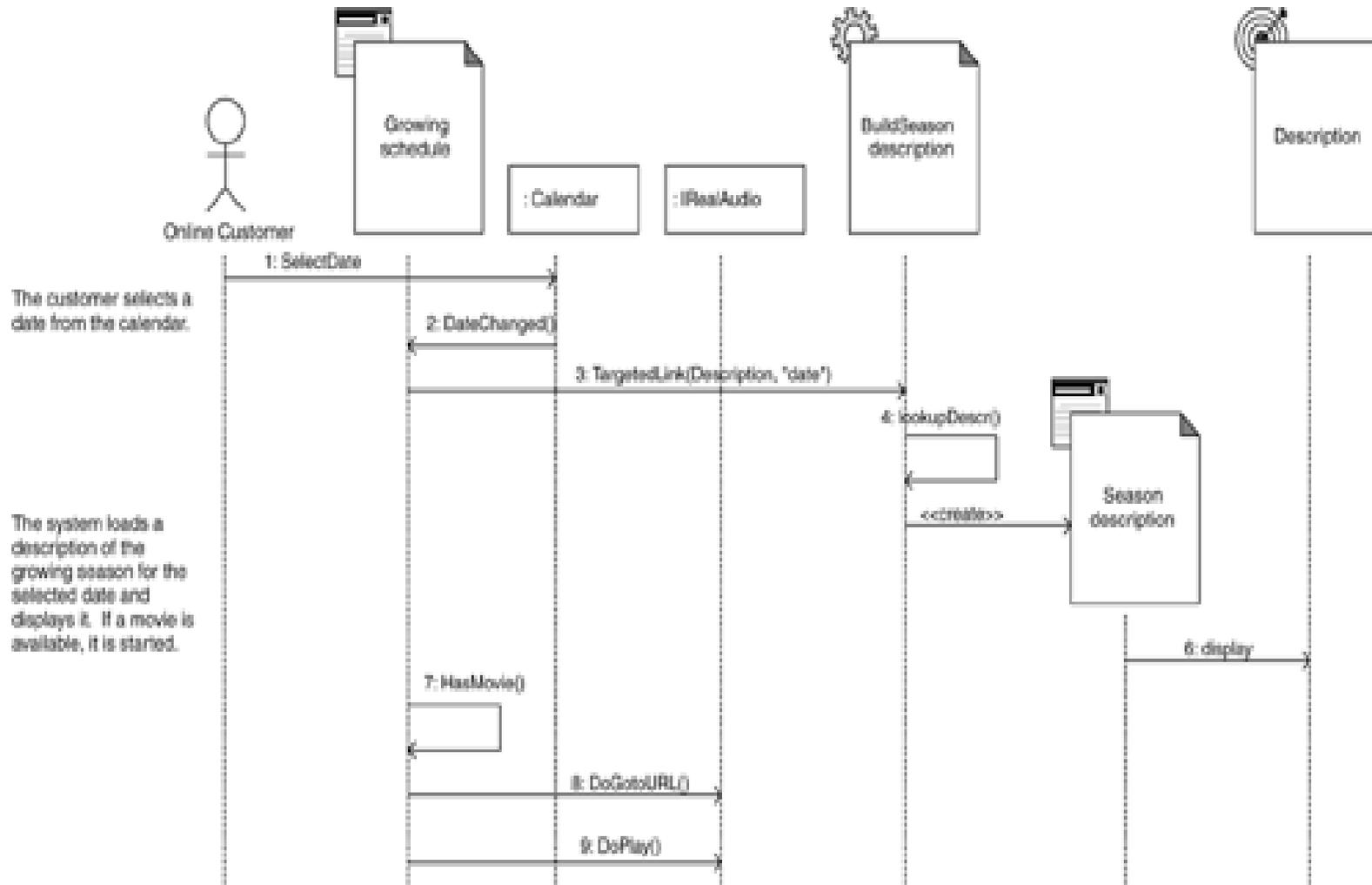


Diagramme de séquence impliquant des composants côté client



La conception et le modèle objets distribués (1)

- **Pour une flexibilité maximale, les applications Web peuvent utiliser des objets réels distribués.**
- **La décision de savoir quand utiliser les applets Java et les beans, RMI / IIOP, ActiveX ou DCOM dépend des besoins de l'application et du niveau d'expérience de l'équipe de développement avec ces technologies.**

La conception et le modèle objets distribués (2)

- la modélisation des applications Web compatibles DCOM est simple.
- Les objets sur le client peuvent interagir avec et envoyer des messages à d'autres objets sur le client (et de même pour les objets serveur), mais la communication entre les objets client et serveur représente quelque chose d'important sur le plan architectural.
- En raison de cette importance, les associations entre les objets s'exécutant sur le client et les objets s'exécutant sur le serveur sont stéréotypées << DCOM >>, en supposant que c'est le mécanisme de communication.

Diagramme de classes avec des objets distribués

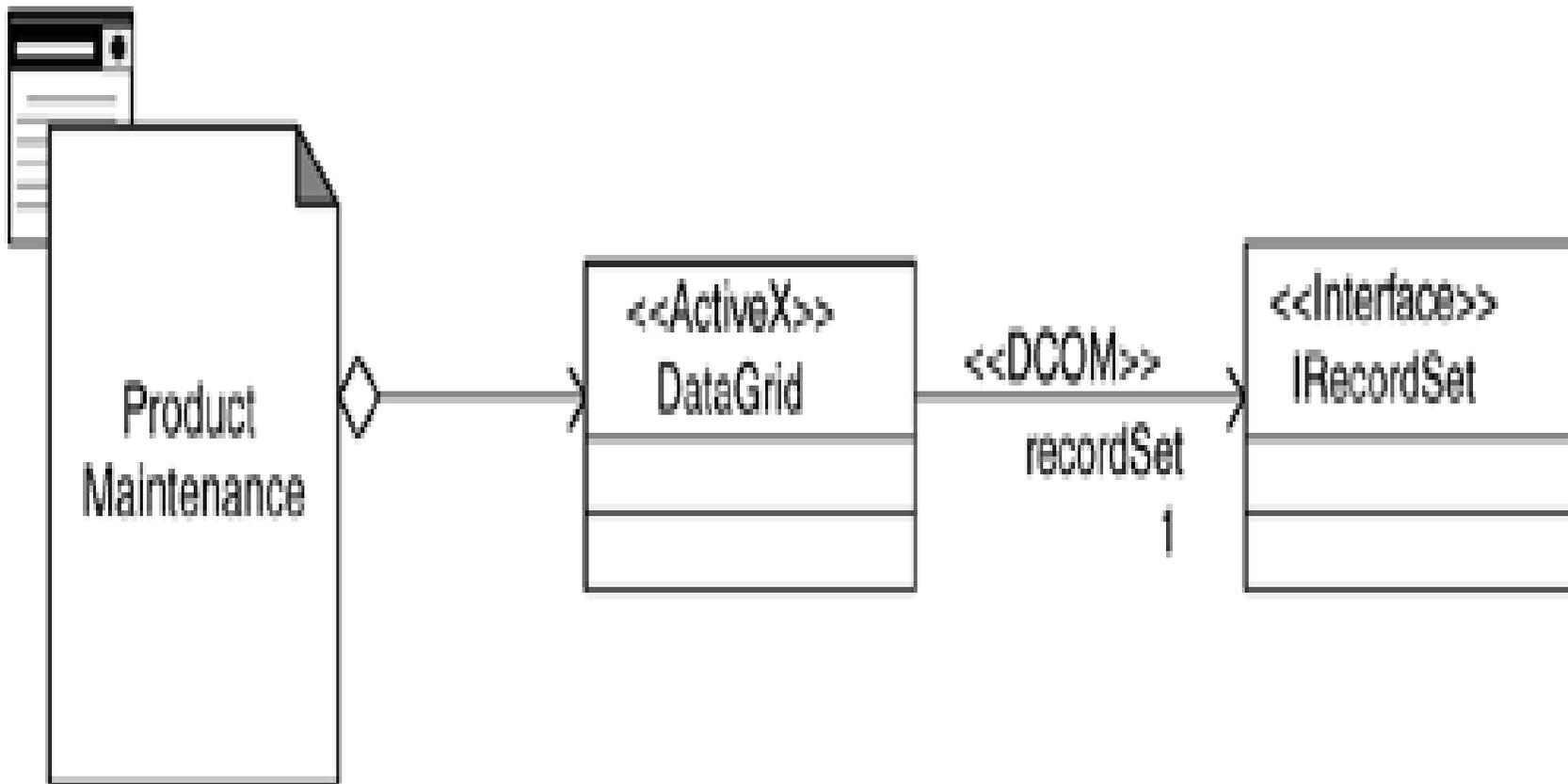
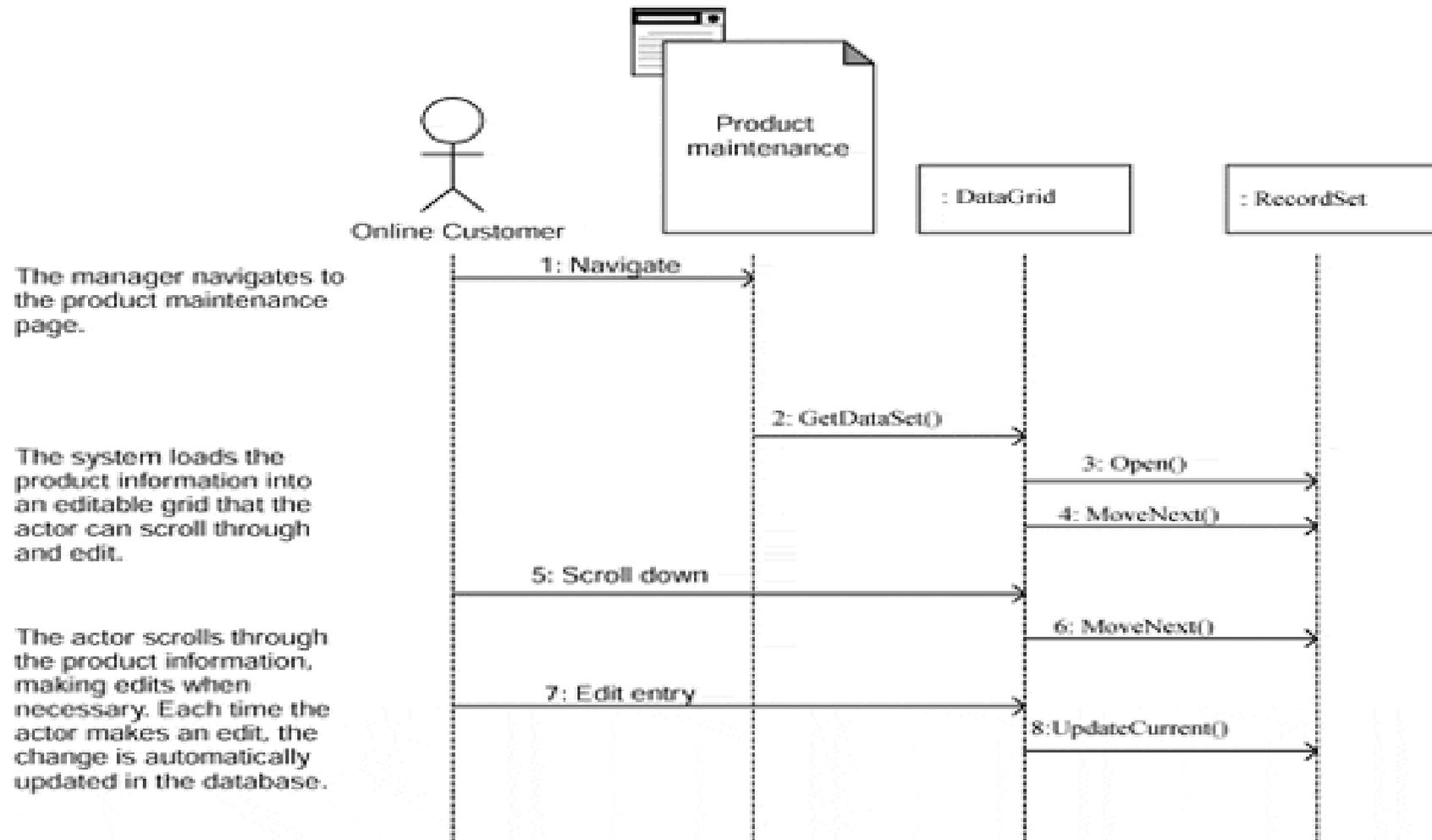
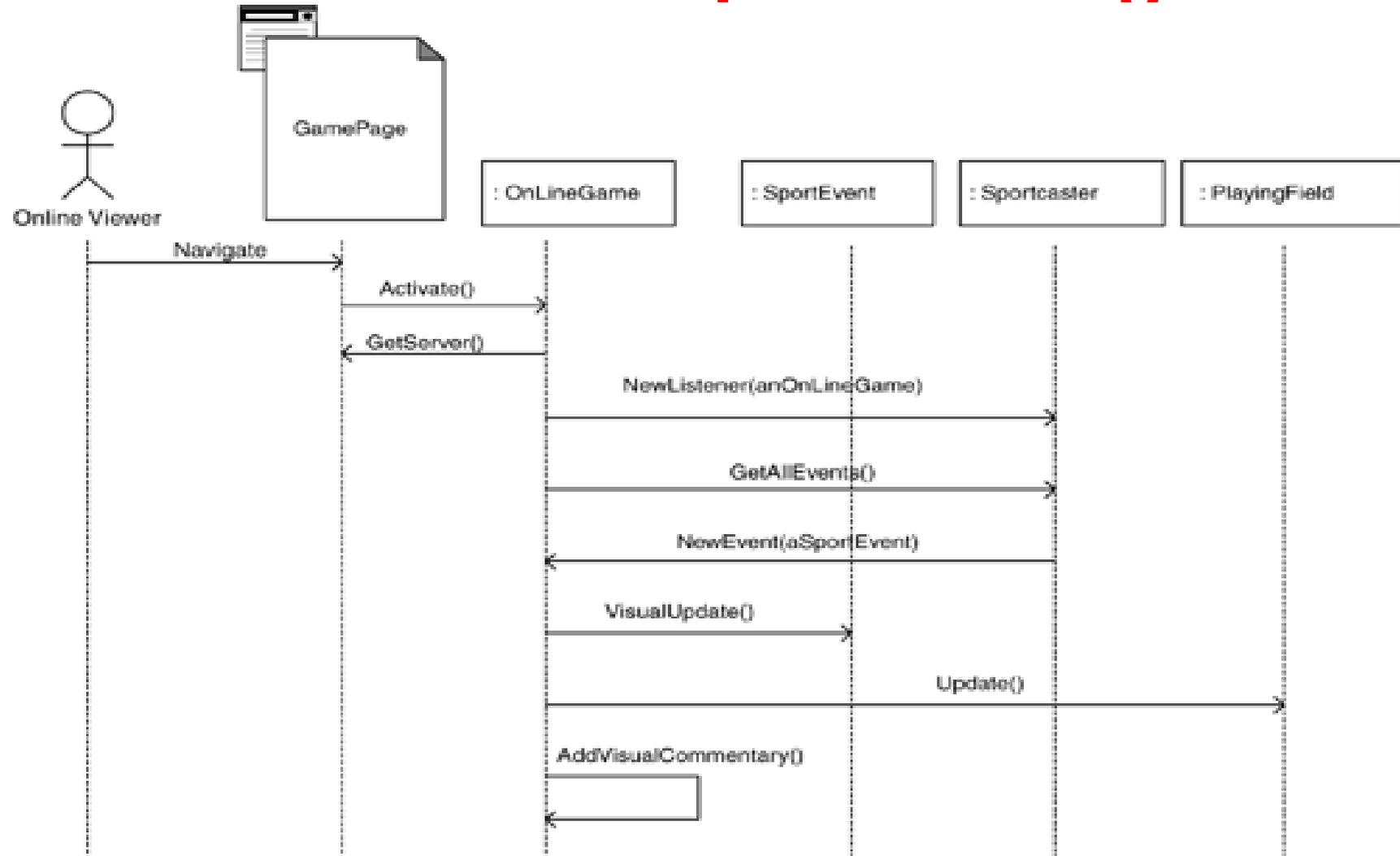


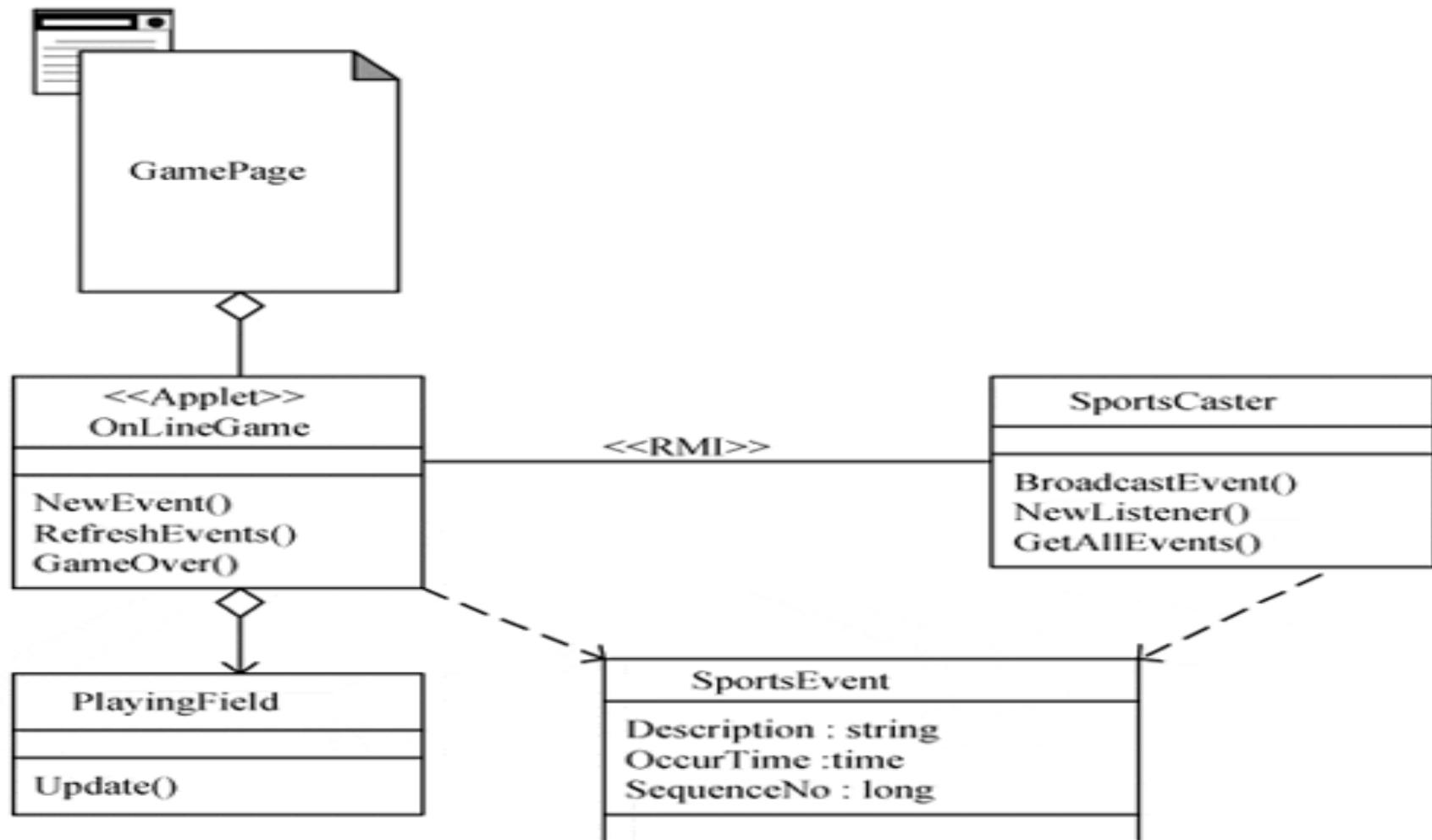
Diagramme de séquence du cas d'utilisation Entrée de produit



Un système de surveillance d'événements sportifs en ligne

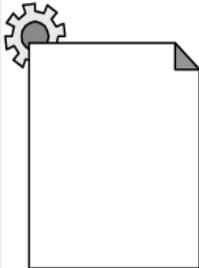


Vue logique de l'application de surveillance d'événements sportifs

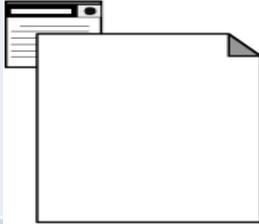


Quelques exemple de stéréotypes

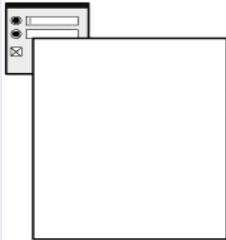
Le stéréotype (page serveur)

Metamodel class	Class
Description	<p>A server page represents a Web page that has scripts executed by the server. These scripts interact with server resources such as databases, business logic, and external systems. The object's operations represent the functions in the script, and its attributes represent the variables that are visible in the page's scope (accessible by all functions in the page).</p>
Icon	
Constraints	<p>Server pages can have relationships only with objects on the server.</p>
Tagged values	<p>Scripting engine: Either the language or the engine that should be used to execute or interpret this page (JavaScript, VBScript, PERL, and so on).</p>

Le stéréotype (page client)

Metamodel class	Class
Description	An instance of a client page is an HTML-formatted Web page and is a mix of data, presentation, and even logic. Client pages are rendered by client browsers and may contain scripts that are interpreted by the browser. Client page functions map to functions in script tags in the page. Client page attributes map to variables declared in the page's script tags that are accessible by any function in the page (page scoped). Client pages can have associations with other client or server pages.
Icon	
Constraints	None
Tagged values	TitleTag: The title of the page as displayed by the browser. BaseTag: The base URL for dereferencing relative URLs. BodyTag: The set of attributes for the <body> tag, which sets background and default text attributes.

Le stéréotype (Form)

Metamodel class	Class
Description	<p>A class stereotyped as a <<form>> is a collection of input fields that are part of a client page. A form class maps directly to the HTML <<form>> tag. The attributes of this class represent the HTML form's input fields (input boxes, text areas, radio buttons, check boxes, and hidden fields). A <<form>> has no operations, since they can't be encapsulated in a form. Any operations that interact with the form are the property of the page that contains the form.</p>
Icon	
Constraints	None
Tagged values	The method—either GET or POST—used to submit data to the action URL.

Conclusion

- L'étape de conception permet de faire la liaison entre l'analyse et l'architecture choisie pour l'application modélisée.
- Les diagrammes réalisés durant cette étape peuvent être mappés directement vers le code