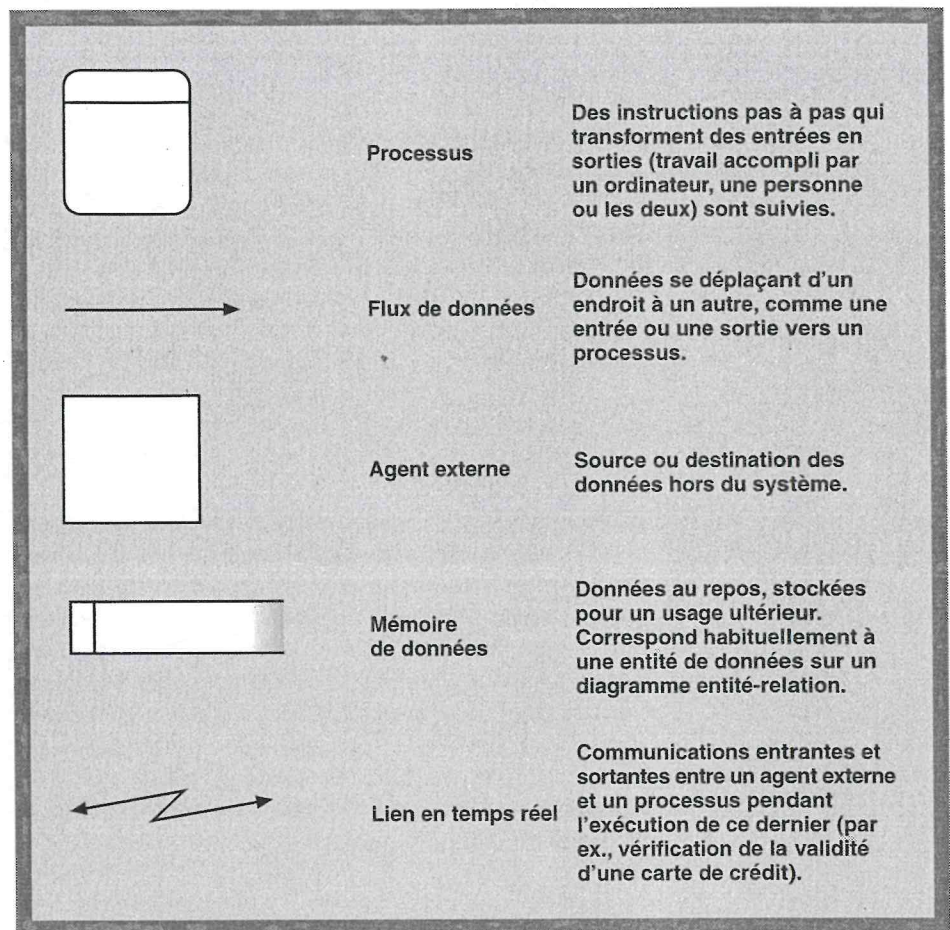


FIGURE 6-2

Symboles du diagramme de flux de données



agent externe

Personne ou organisation hors des limites du système qui fournit des entrées de données ou accepte des sorties de données.

processus

Sur un DFD, symbole qui représente une procédure ou un algorithme par lequel des entrées de données sont transformées en sorties de données.

flux de données

Sur un DFD, flèche qui désigne le déplacement des données à travers des processus, des mémoires de données et des agents externes.

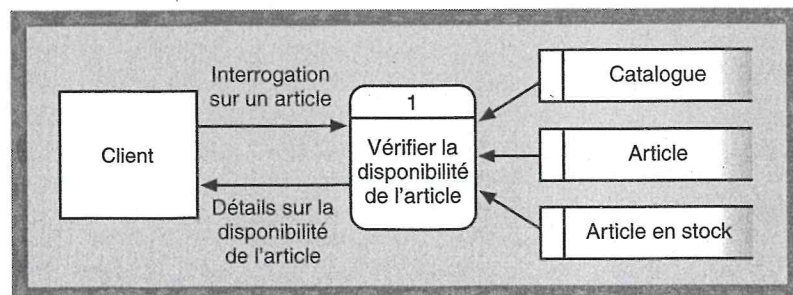
mémoire de données

Endroit où des données sont conservées, en attente d'accès futurs par un ou plusieurs processus.

FIGURE 6-3

DFD illustrant le processus « Vérifier la disponibilité d'un article » (un fragment de DFD du cas RMO)

La figure 6-3 illustre un exemple de diagramme de flux de données représentant une partie du système de soutien des clients de *Rocky Mountain Outfitters* (RMO). Le carré désigne un *agent externe* (Client), soit la source et la destination de certaines données hors du système. Le rectangle aux coins arrondis est un *processus* nommé *Vérifier la disponibilité de l'article*, auquel on peut aussi se référer par son numéro (le 1). Un processus définit les règles de transformation des entrées en sorties. Les lignes munies de pointes de flèches représentent les *flux de données*. La figure 6-3 montre deux flux de données entre le Client et le processus 1 : l'entrée au processus appelée *Interrogation sur un article* et la sortie du processus nommée *Détails sur la disponibilité de l'article*. Le symbole de fin, soit un rectangle plat à trois côtés, désigne une *mémoire de données*. Chaque mémoire de données représente un fichier ou une partie d'une base de données stockant des informations sur une entité de données. Dans cet exemple, les flux de données (les flèches) pointent depuis les mémoires de données vers le processus, ce qui signifie que le processus cherche de l'information dans les mémoires de données Catalogue, Article et Article en stock.

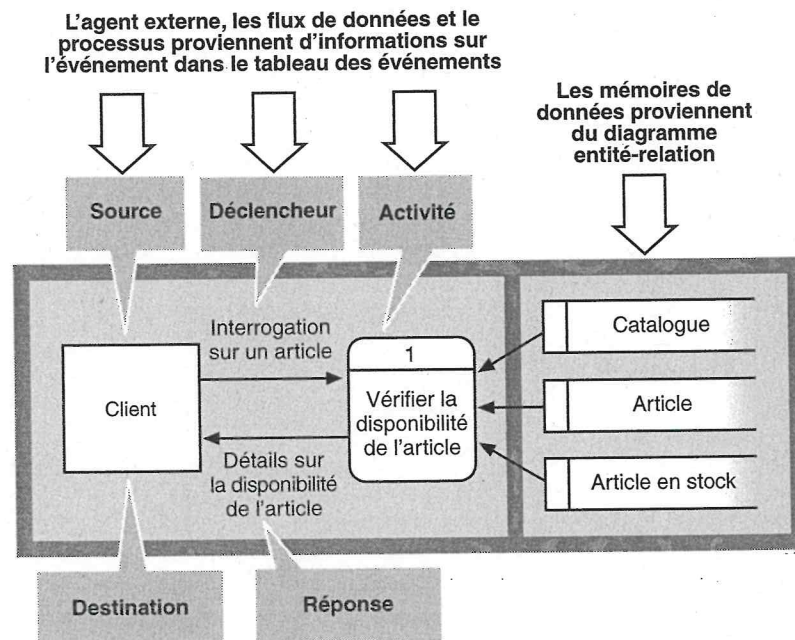


Vous avez peut-être remarqué que le processus montré à la figure 6-3 correspond à une activité du tableau des événements de RMO illustré à la figure 5-15 du chapitre 5. L'événement correspondait à *Le client veut vérifier la disponibilité d'un article*, le déclencheur était *Interrogation sur un article*, la source était *Client*, la réponse était *Détails sur la disponibilité de l'article* et la destination de la réponse correspondait à *Client*. Le diagramme de flux de données illustre donc sous forme de graphique l'activité du système en réponse à cet événement particulier.

Certains éléments d'information du DFD n'apparaissent toutefois pas dans le tableau des événements tels que les mémoires de données qui stockent les informations sur la disponibilité d'un article. À la figure 6-3, chaque mémoire de données représente une entité de données tirée du diagramme entité-relation (ERD) présenté à la figure 5-28 du chapitre 5. Sur le DFD, le processus utilise les informations que nous avons fournies en incluant ces entités de données et leurs attributs dans l'ERD pour le système. Le diagramme de flux de données intègre donc les processus déclenchés par les événements avec les entités de données modélisées au moyen de l'ERD. La figure 6-4 résume les correspondances entre les composants du DFD, les événements du tableau des événements et les entités définies dans l'ERD.

FIGURE 6-4

DFD intègre le tableau des événements et l'ERD.



Diagrammes de flux de données (DFD) et niveaux d'abstraction

De nombreux types de diagrammes de flux de données sont produits pour illustrer les spécifications d'un système. L'exemple précédent est un fragment de DFD, qui montre un processus en réponse à un événement. D'autres diagrammes de flux de données présentent le processus soit à un niveau supérieur (une vue plus générale du système), soit à un niveau inférieur (une vue plus détaillée d'un processus). Ces différentes vues d'un système (niveau supérieur et niveau inférieur) se nomment *niveaux d'abstraction*.

La capacité du diagramme de flux de données à illustrer des niveaux supérieurs ou inférieurs d'un système est une autre caractéristique très utile. Sur un DFD, il est possible de décomposer les processus de niveau supérieur en DFD détaillés et distincts de niveau inférieur. On peut aussi décomposer les processus de DFD détaillés en diagrammes supplémentaires pour obtenir de multiples niveaux d'abstraction.

La figure 6-5 illustre comment des DFD à chaque niveau de détail fournissent des informations additionnelles sur un processus dont le niveau est supérieur d'un rang. Le DFD du haut montre la représentation la plus abstraite d'un système d'inscription à des cours, en tant que processus unique. Le DFD central illustre les détails internes d'un processus de diagramme contextuel. Quant au DFD du bas, il montre les détails intrinsèques au processus 1 du DFD central. Les sections suivantes analysent plus en profondeur chaque niveau d'abstraction de DFD.

Niveaux d'abstraction

Une technique de modélisation qui divise un système en une hiérarchie de modèles plus en plus détaillés.

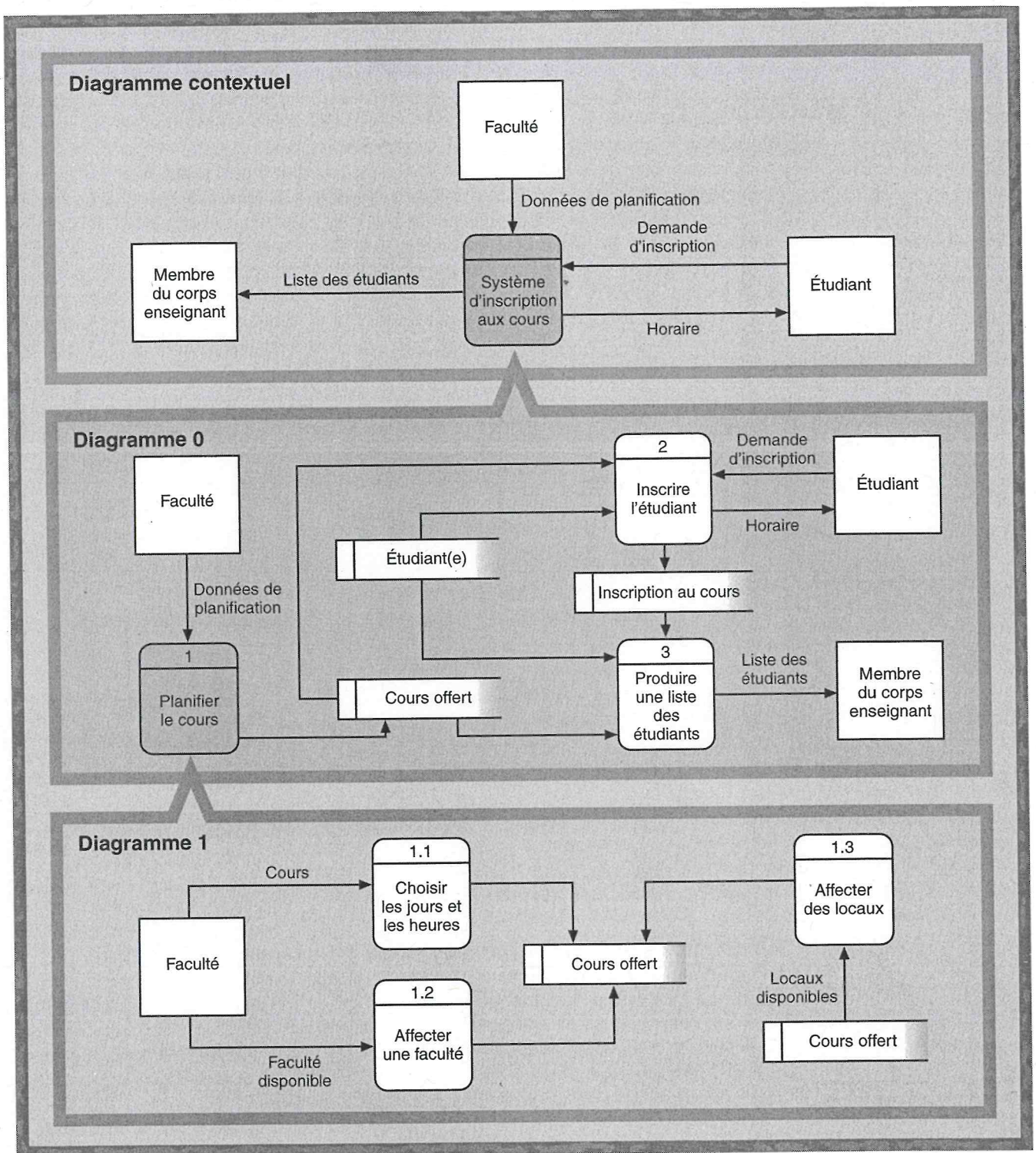


FIGURE 6-5

Niveaux d'abstraction de DFD pour un système d'inscription à des cours

diagramme contextuel

DFD qui résume toutes les activités de traitement à l'intérieur d'un système en un unique symbole de processus.

Diagramme contextuel

Un *diagramme contextuel* est un DFD qui décrit la vue la plus abstraite d'un système. Tous les agents externes et flux de données entrants et sortants du système sont regroupés dans un seul diagramme où le système complet est représenté comme un processus unique. À la figure 6-5, le DFD du haut montre le diagramme contextuel d'un système d'inscription à des cours qui interagit avec trois agents externes: la Faculté, l'Étudiant et le Membre du corps enseignant. Les facultés fournissent les informations sur les cours offerts, les étudiants font des demandes d'inscription aux cours proposés et les membres du corps enseignant reçoivent les listes d'étudiants une fois la période d'inscription terminée.

Les diagrammes contextuels sont utiles pour illustrer les limites d'un système. On définit la portée d'un système par ce qui est décrit dans le processus unique et par ce qui est représenté comme agents externes. Les agents externes qui fournissent ou reçoivent des données du système sont hors de la portée du système, tandis que tout le reste est à l'intérieur de cette portée. D'ordinaire, le diagramme contextuel n'illustre pas les mémoires de données puisque celles-ci sont considérées comme incluses dans la portée du système (comme si elles faisaient partie de l'implémentation interne du processus représentant le système). On peut toutefois illustrer les mémoires de données lorsqu'elles sont partagées à la fois par le système en cours de modélisation et un autre système.

FIGURE 6-6

Diagramme contextuel pour le système de soutien des clients de RMO

La figure 6-6 montre un diagramme contextuel pour le système de soutien des clients de *Rocky Mountain Outfitters*. D'habitude les flux de données et agents externes du diagramme contextuel proviennent directement du tableau des événements. Toutefois, comme le système de soutien des clients de RMO répond à 20 événements, cette figure fusionne les flux de données de certains événements à des fins de simplification.

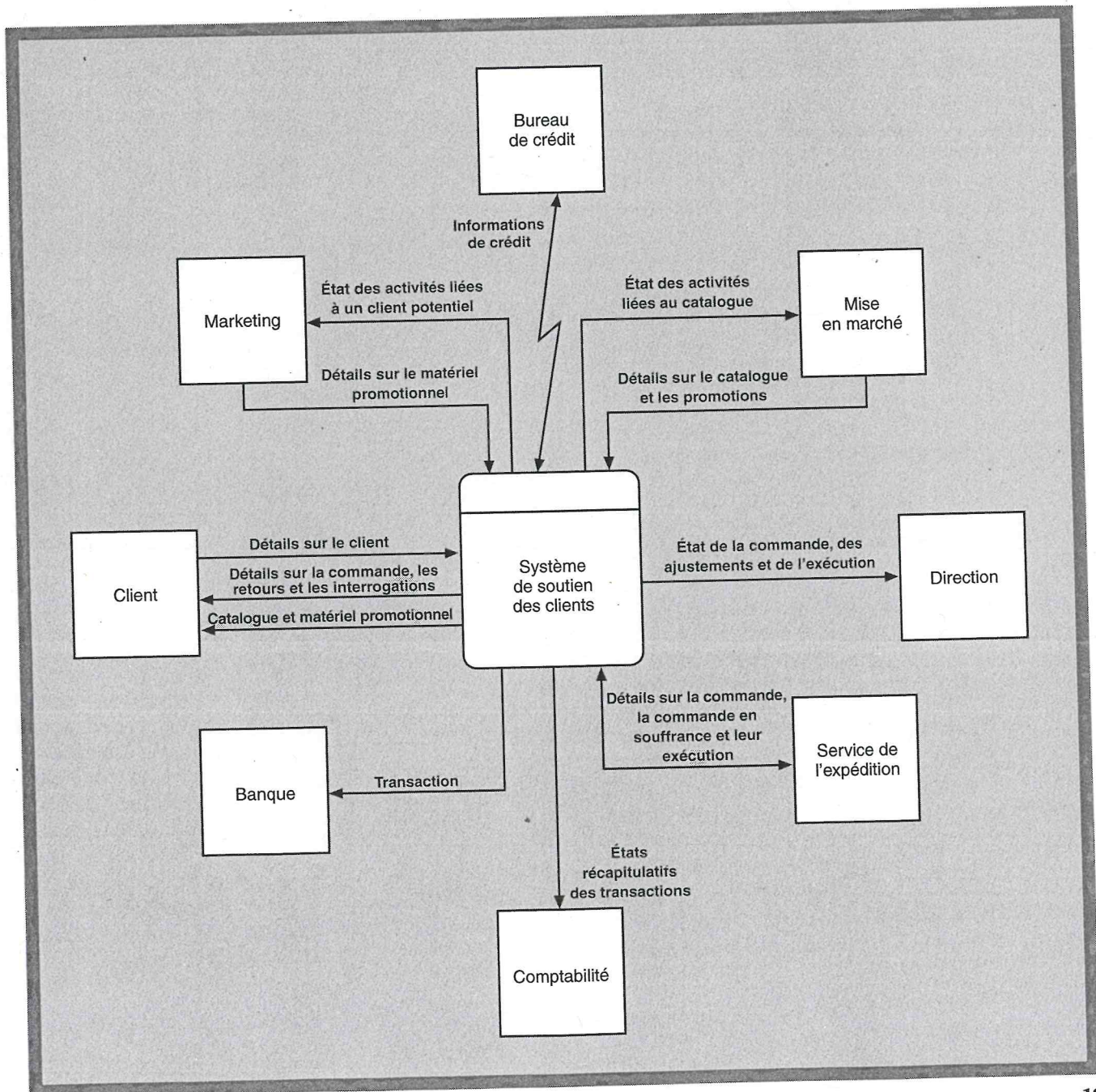


FIGURE 6-7

Sous-systèmes de RMO et leurs événements

Lorsqu'un système répond à de nombreux événements, il est généralement divisé en sous-systèmes et il faut alors créer un diagramme contextuel pour chacun. La figure 6-7 subdivise le système de soutien des clients de RMO en fonction des similitudes entre les événements, y compris les interactions avec les agents externes, les interactions avec les mémoires de données et les similarités dans les processus requis. La figure 6-8 illustre le diagramme contextuel du sous-système d'entrée des commandes.

Sous-système d'entrée des commandes

- Le client veut vérifier la disponibilité d'un article.
- Le client passe une commande.
- Le client modifie ou annule une commande.
- Moment de produire les états récapitulatifs des commandes.
- Moment de produire les états récapitulatifs des transactions.

Sous-système d'exécution des commandes

- Le client ou la direction veut vérifier le statut d'une commande.
- Le service de l'expédition traite la commande.
- Le service de l'expédition identifie une commande en souffrance.
- Le client renvoie un article (l'article est défectueux, le client a changé d'avis, retour complet ou partiel).
- Moment de produire les états récapitulatifs d'exécution des commandes.

Sous-système de maintenance des clients

- Un client potentiel demande un catalogue.
- Moment de produire les rapports sur les activités liées aux clients potentiels.
- Informations sur les mises à jour des comptes clients.
- Le service du marketing veut envoyer du matériel promotionnel aux clients.
- La direction ajuste les frais aux clients (corrige les erreurs, accorde des privilèges).
- Moment de produire les états d'ajustements ou de privilèges.

Sous-système de maintenance du catalogue

- Le service de la mise en marché met le catalogue à jour (ajouts, modifications, suppressions ou changements de prix).
- Le service de la mise en marché crée une promotion spéciale.
- Le service de la mise en marché produit un nouveau catalogue.
- Moment de produire les états sur les activités liées au catalogue.

fragment de DFD

DFD qui représente la réponse du système à un événement à l'intérieur d'un unique symbole de processus.

Fragments de DFD

Un *fragment de DFD* est créé pour chaque événement d'une liste d'événements (développée dans le tableau des événements). Chaque fragment de DFD est un modèle indépendant qui illustre comment le système répond à un événement donné. Habituellement, l'analyste crée les fragments de DFD un par un, en concentrant son attention sur chaque partie du système.

La figure 6-9 illustre les trois fragments de DFD pour le système d'inscription à des cours. Chaque fragment de DFD représente l'ensemble du processus requis pour un événement à l'intérieur d'un unique symbole de processus. Par contre, les fragments montrent les détails des interactions entre le processus, les agents externes et les mémoires de données internes. Les mémoires de données utilisées sur un fragment de DFD représentent les entités du diagramme entité-relation. Chaque fragment de DFD n'illustre que les mémoires de données nécessaires pour répondre à l'événement.

FIGURE 6-8

Diagramme contextuel pour
le sous-système d'entrée des
commandes de RMO

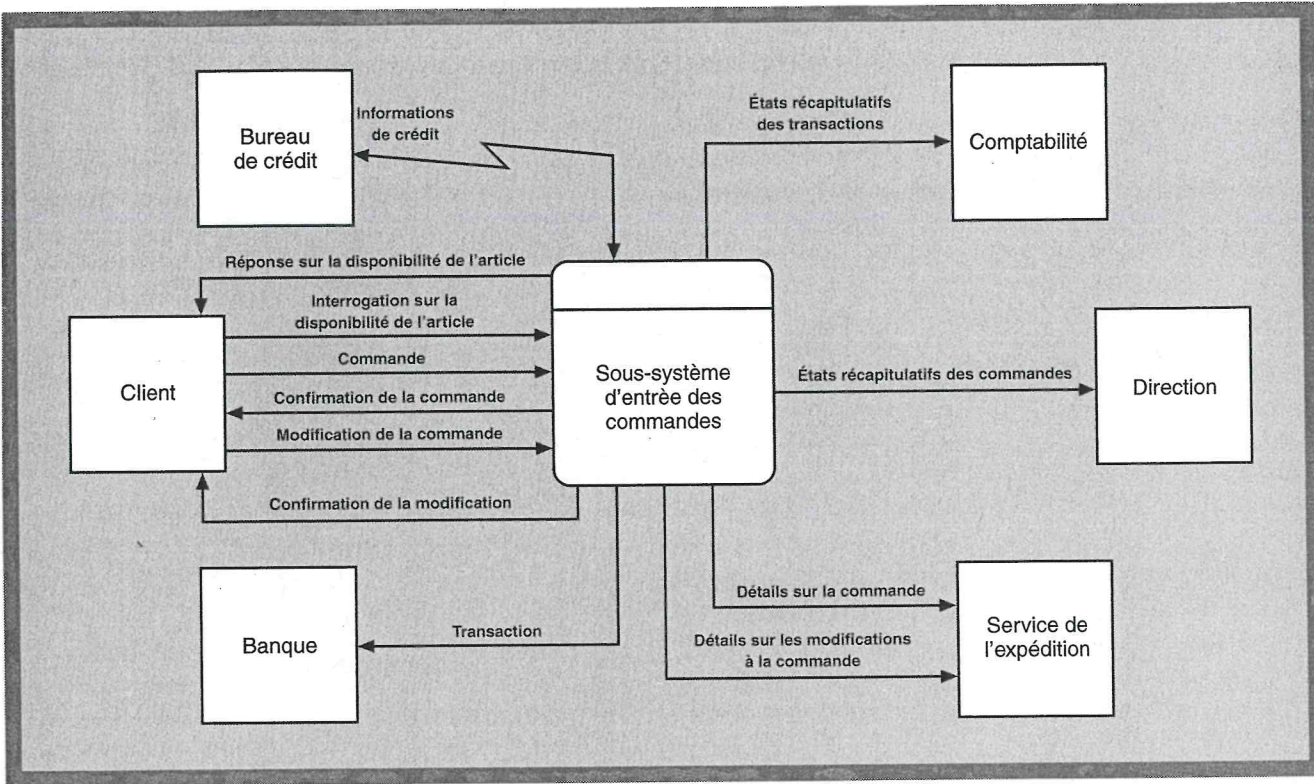
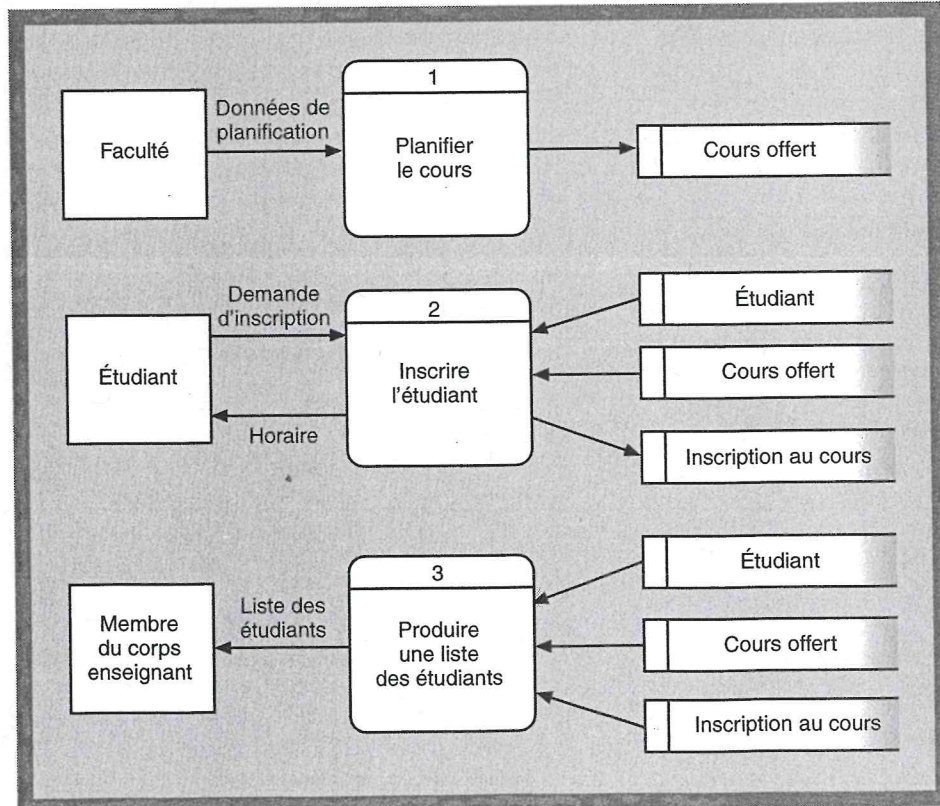


FIGURE 6-9

Diagrammes de DFD pour
le système d'inscription à
des cours



Le diagramme 0 sert essentiellement d'outil de présentation. Il permet de résumer un système entier ou un sous-système de façon plus détaillée qu'un diagramme contextuel. On peut aussi l'utiliser lors de la conception de bases de données, car il illustre tous les flux de données entrant et sortant des mémoires de données. Néanmoins, les analystes évitent souvent de développer le diagramme 0 et ce pour deux raisons :

- Le contenu des informations duplique la série des fragments de DFD.
- Le diagramme est souvent complexe et peu maniable, surtout dans le cas de gros systèmes répondant à de nombreux événements.

Nous verrons d'ailleurs plus loin dans ce chapitre que la redondance et la complexité sont deux caractéristiques des DFD que les analystes doivent éviter le plus souvent possible.

Décomposition pour visualiser les détails d'une activité

Certains fragments de DFD impliquent des charges de traitement considérables que l'analyste se doit d'explorer plus en détail. Comme toute autre étape de modélisation, décomposer davantage aide l'analyste à en apprendre plus sur les spécifications tout en produisant la documentation requise. La figure 6-13 illustre l'exemple d'un diagramme plus détaillé pour le fragment 2 du DFD de RMO, *Créer une nouvelle commande*. Ce fragment est nommé diagramme 2 puisqu'il comprend les « entrailles » du processus 2. Les sous-processus sont numérotés 2.1, 2.2, 2.3 et 2.4. Ce système de numérotation ne représente toutefois pas nécessairement leur ordre d'exécution.

FIGURE 6-13

Diagramme détaillé pour
Créer une nouvelle commande
(fragment 2)

