

GSI

Gestion des systèmes d'information

Atelier C : Évolution du SI et développement des applications

Avec le souci d'éclairer les choix pédagogiques et didactiques préconisés par le programme, trois exemples de séquences nous ont été proposés s'appuyant sur les trois notions suivantes :

- Les fonctions et procédures (intervention de Patrice Grand),
- Les jeux d'enregistrements (intervention de Christine Gaubert-Macon),
- Les cas d'utilisation (travail par groupes à partir de documents fournis s'appuyant sur les contextes de systèmes d'information publiés par le CERTA).

Rappel : Toutes les notions abordées devront l'être en s'appuyant sur les ressources fournies par les contextes.

Première partie de l'atelier : les fonctions et procédures

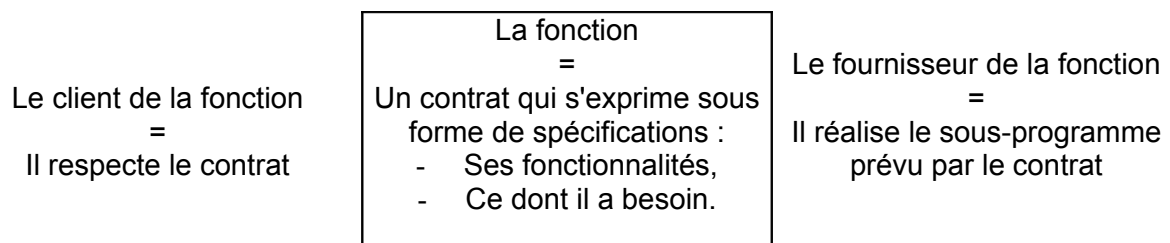
Patrice Grand nous a proposé une séquence pédagogique articulée en quatre séances. Un document de 15 pages, expose de façon détaillée ces 4 séances ainsi que les choix didactiques opérés.

Pour amener l'élève à maîtriser les notions du programme liées aux procédures et aux fonctions, il est important de lui faire endosser les trois rôles successifs suivants :

- Rôle de l'utilisateur final de l'application,
- Rôle du développeur qui utilise la fonction pour développer l'application (dans ce cas, l'élève joue le rôle du **client de fonction**),
- Rôle du développeur de fonction (dans ce cas, l'élève est le **fournisseur de fonction**).

Les collègues ayant participé à l'atelier C ont estimé que se positionner par rapport à ces 3 rôles distincts peut présenter des difficultés pour l'élève. Il faudra donc varier les situations pédagogiques pour atteindre les objectifs.

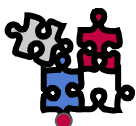
La fonction peut alors être abordée comme un **contrat**.



Il est important de faire utiliser des fonctions dès l'utilisation contexte de productivité individuelle (contexte *GestProj*), en **insistant sur la notion de contrat**.

Ensuite, on pourra utiliser les contextes suivants pour aménager des activités de réinvestissement et de consolidation.

Bornage :



GSI

La présentation de Patrice et les discussions qui ont suivies nous ont permis de mieux border le programme :

- Il n'est pas indispensable de rentrer dans la différence entre fonction et procédure qui est d'ailleurs contingente au langage utilisé. Le seul terme de fonction a donc été choisi et les exemples présentés sont des fonctions qui retournent une valeur.
- La notion de procédure pourra être vue comme une fonction particulière qui ne contient pas de variable de retour. Il est opportun de privilégier les fonctions par rapport aux procédures.
- Il est inutile de faire la différence entre passage par valeur et passage par adresse/référence : on utilisera le mode de passage par défaut du langage de programmation utilisé.
- Il faut se limiter à des fonctions dont le nombre de paramètres est fixe et donc tous obligatoires. C'est pourquoi d'ailleurs l'exemple de la fonction SOMME() dans EXCEL utilisée dans la séance 1 n'est pas apparu comme le plus pertinent.
- Il faut éviter l'écriture de fonctions hors contexte.

Remarques :

Cette notion a normalement été vue en Première mais il est prudent de reprendre les choses à la base.

Le contexte de système d'information *GestProj*¹ est suffisamment riche pour introduire les trois premières séances qui ont essentiellement pour but de mettre l'élève dans la position d'un utilisateur de fonctions préexistantes pour comprendre

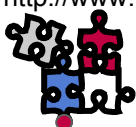
- Comment choisir une fonction par rapport au résultat attendu et comment l'utiliser correctement,
- La notion de paramètre(s) et de valeur de retour,
- Le mécanisme d'imbrication de fonctions.

La séance 4 est basée sur le contexte de gestion des formations² et met l'élève dans la posture d'un fournisseur de fonction. À travers la traçabilité du code VBA d'ACCESS l'élève doit comprendre :

- La notion d'appel d'une fonction et le « décrochage » de code qu'elle induit,
- La notion de variable interne (locale) à la fonction.

¹ <http://www.reseaucerta.org/cotecours/pub.php?num=364>

² <http://www.reseaucerta.org/cotecours/pub.php?num=362>



Seconde partie de l'atelier : les jeux d'enregistrements

Christine Gaubert Macon a présenté 3 séances d'une durée totale de 8h (2 + 2 + 4) pour comprendre le rôle de cet objet technique et son fonctionnement en étudiant la façon dont il est exploité dans une programmation PHP.

L'ensemble des participants a adhéré au bornage proposé :

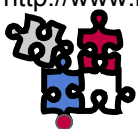
- Parcours avant de tout le jeu. Lorsqu'on veut modifier les résultats fournis dans un jeu d'enregistrements, on modifie la requête ayant permis de l'obtenir.
- Fourniture d'un squelette de programme de parcours d'un jeu d'enregistrements.

La séance proposée s'articulait autour de trois séquences basées sur l'exploitation du contexte *Festival*³. Un document de douze pages détaille la démarche proposée. Cette séance devrait être complétée par une mise en situation où le jeu d'enregistrements ne renvoie qu'une seule ligne, ce qui nécessite de revoir l'algorithme-type de traitement du jeu d'enregistrements tel qu'il est fourni dans le squelette de programme.

Il est apparu important de trouver des situations pédagogiques d'observation permettant aux élèves de comprendre que le jeu d'enregistrements, lorsqu'il est exploité, l'est ligne par ligne, ceci dans le but de faire émerger la notion de curseur. Un modèle du jeu d'enregistrements a été proposé comme concept à enseigner.

On peut aussi, pour appréhender la notion de jeu d'enregistrements, modifier à sa guise le code source fourni par le contexte *Gestion des formations* afin de créer une situation permettant à l'élève d'y manipuler un jeu d'enregistrements.

³ <http://www.reseaucerta.org/cotecours/pub.php?num=390>



Troisième partie de l'atelier : les cas d'utilisation

Cette notion, nouvelle pour la plupart des participants, en tout cas dans sa dimension pédagogique, a permis aux collègues répartis dans trois groupes de réfléchir et de proposer éventuellement des pistes de scénarios pédagogiques possibles, l'objectif étant d'amener l'élève à comprendre, savoir utiliser et modifier des cas d'utilisation en s'appuyant sur les 3 contextes de SI.

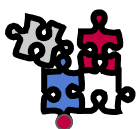
Chaque groupe de travail disposait de la description des cas d'utilisation des trois contextes ainsi qu'un document de cadrage fourni en annexe 1.

Le contexte *GestProj* est apparu trop pauvre. On n'aborde donc cette notion qu'au moment du contexte *Gestion des formations*.

- Une première séance d'observation de la fiche descriptive d'un cas d'utilisation permet aux élèves de découvrir le vocabulaire lié à cette notion. Si les concepts d'action, d'événement déclencheur et de scénario ne semblent pas poser de problème, en revanche celles de parties prenantes et d'intérêts, de portée et de pré-conditions n'apparaissent pas d'emblée comme très claires. On pourrait donc proposer un formalisme plus minimal qui masquerait ces concepts d'autant plus qu'il est exclu d'aborder les relations entre les cas d'utilisation. En revanche la transversalité avec la représentation graphique des processus est opportune.
- Dans une deuxième séance, l'élève est placé dans la situation d'utiliser un formulaire avec un jeu d'essai suffisamment fourni pour envisager tous les cas de figure, y compris les cas particuliers qui engendrent des dysfonctionnements. Il est ensuite invité à vérifier si la fiche descriptive du cas est conforme ou non avec ce qu'il a observé.
- Dans une troisième séance on pourrait demander à l'élève, à partir d'un formulaire, de concevoir lui-même un jeu d'essai puis de rédiger la fiche descriptive d'un cas d'utilisation.
- Dans une quatrième séance l'élève devrait répondre à une amélioration d'un formulaire existant en modifiant la fiche descriptive correspondante ...

Il faut être vigilant sur deux points :

- Varier suffisamment les situations pédagogiques de manière à permettre à l'élève de bien situer le cas d'utilisation dans la démarche de conception d'une application (le cas d'utilisation **préexiste** à l'application),
- Attention au risque de confusion entre **scénario** et **mode opératoire de l'application**. Pour prévenir ce risque, différents moyens pédagogiques existent, comme par exemple le fait de fournir en même temps le scénario et le mode opératoire de l'application, ou bien encore en s'appuyant sur l'expression des besoins via un compte-rendu d'entretien.



Conclusion générale

En mêlant habilement pédagogie inductive et déductive il est possible d'aborder ces trois notions en s'appuyant à chaque fois sur la richesse des trois contextes proposés.

Le volume de ce qui nous a été proposé dans l'atelier C constitue un bon nombre de pistes de travail.

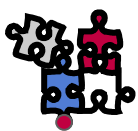
Il faut perpétuellement procéder à des allers et retours entre manipulation et formalisation, cette dernière ne devant surtout pas être négligée.

En règle générale, il faut rester vigilant à ne pas construire des formalismes dans le but d'aider à la compréhension, mais pour finalement compliquer l'appropriation des notions par les élèves.

Il apparaît opportun d'introduire les notions dès que le contexte étudié le permet, ce qui donne la possibilité de "repasser" plusieurs fois sur les notions à faire acquérir aux élèves.

Enfin, il ne faut pas hésiter à modifier un contexte ou ses supports pour mieux servir les objectifs pédagogiques visés.

Remarque : l'annexe 2 de ce document présente des pistes de réflexion sur le thème de la prise en charge du dialogue homme-machine. La durée de l'atelier n'a pas permis de travailler sur cette notion du programme.



Annexe 1 : Document de cadrage pour le travail en groupe sur les cas d'utilisation.

Construction d'une séquence sur le thème « Formalisation des besoins »

○ Problématique

Une application est un logiciel qui doit répondre aux besoins métier d'utilisateurs. La formalisation des besoins fait partie de la première étape dans le cycle de vie de l'application.

L'application est testée en prenant appui sur les cas d'utilisation. Ce qui renforce l'importance d'une définition soignée.

○ Référence au programme

1.2. Formalisation des besoins	L'expression des besoins permet de définir ce qui est attendu de l'application. Ces besoins sont formalisés : - par un diagramme de cas d'utilisation qui présente les services attendus ; - par des jeux d'essai destinés à vérifier la conformité de la solution aux attentes.	- Expression des besoins : cas d'utilisation, jeu d'essai.
--------------------------------	--	--

1.2. Formalisation des besoins

En classe terminale, les besoins sont exprimés sous forme de cas d'utilisation schématisés à l'aide d'un diagramme détaillant les interactions entre l'application et les utilisateurs. Chaque cas d'utilisation décrit un service rendu par l'application à un ou plusieurs utilisateurs. Il s'accompagne d'une description textuelle qui décrit le scénario le plus courant (scénario nominal) ainsi que les variantes (traitement des exceptions, comportements alternatifs).

Pour faire évoluer l'application existante, l'élève dispose :

- du schéma de représentation du processus correspondant qui lui permet de percevoir sur quelle partie de l'organisation, circonscrite à une activité, il doit agir ;
- d'une présentation de l'application (fonctionnalités, types d'utilisateurs et contraintes d'exécution) ;
- de la description des cas d'utilisation de l'application existante et ceux de la future application ;
- de l'application elle-même avec son exécutable et son code source, de ses modes opératoires et de sa documentation ;
- de la base de données utilisée par l'application et du SGBD associé ;
- du logiciel de développement qui a permis la création de l'application.

L'analyse de l'application existante et la description des cas d'utilisation fournis pour la future application conduisent l'élève à compléter ou rédiger les jeux d'essai qui permettent de s'assurer de la conformité de l'application aux besoins.

○ Aspects didactiques

Objectifs

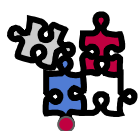
- Comprendre la « grammaire » des cas d'utilisation : acteur, scénario typique, cas particulier(s), étape
- Evaluer l'adéquation entre la définition d'un cas d'utilisation et sa réalisation dans l'application
- Faire évoluer la description d'un cas d'utilisation

Limites

- pas de spécification des relations entre cas d'utilisation
- l'évolution demandée sera relative soit à un cas d'utilisation déjà fourni pour lequel une nouvelle interface aura été spécifiée soit à la définition d'un cas d'utilisation similaire à un cas d'utilisation déjà fourni.

○ Documents fournis

- Les cas d'utilisation des trois applications
- Les URL fournies dans la bibliographie.



Annexe 2 : éléments de réflexion pour une séquence sur le thème « prise en charge du dialogue homme-machine »

○ Problématique

Toute application informatique a pour partie visible par l'utilisateur une interface homme-machine. Son rôle est de dialoguer avec l'utilisateur pour l'exécution de l'application. Elle permet d'une part de présenter des données extraites d'une base de données ou des informations calculées, et d'autre part, pour l'utilisateur, d'agir via des contrôles graphiques, que l'application soit en mode web ou de bureau.

○ Référence au programme

2.1. Prise en charge du dialogue homme-machine	Le dialogue homme-machine permet à l'utilisateur d'interagir avec l'application. L'interface homme-machine (IHM) doit respecter certains principes ergonomiques pour garantir la qualité du dialogue. La réalisation de maquettes, en collaboration avec les utilisateurs, permet de concevoir l'IHM. Les IHM graphiques réagissent à des événements provoqués par des actions de l'utilisateur sur les contrôles graphiques qui les composent.	- Interface homme-machine. - Maquette. - Ergonomie. - Contrôle graphique. - Événement.
--	--	--

2.1. Prise en charge du dialogue homme-machine

Il s'agit d'amener les élèves à identifier et à évaluer les principaux critères ergonomiques auxquels les interfaces utilisateurs doivent répondre : flexibilité, transparence, facilité d'apprentissage, homogénéité. Pour définir une IHM, il s'agit non seulement de décrire les éléments graphiques qui la composent mais aussi de spécifier le comportement de cette interface. L'étude d'une maquette permet de rendre compte de l'aspect dynamique de l'interface.

Les caractéristiques principales des contrôles graphiques les plus utilisés seront étudiées par la présentation de leurs propriétés et de leurs méthodes (fenêtre, bouton, zone de texte, liste, bouton radio, case à cocher, contrôle d'accès aux données). Les élèves doivent être en mesure de choisir ceux qui répondent aux besoins.

○ Aspects didactiques

Objectifs

- connaître les caractéristiques des contrôles graphiques
- justifier le choix des contrôles utilisés dans l'interface (page ou fenêtre)
- justifier ou identifier l'événement sur lequel le contrôle va réagir de manière pertinente (en fonction des besoins)
- faire évoluer une interface pour laquelle les flux d'informations sont fournis.

Limites

- expression des besoins : les flux d'informations attendus sont spécifiés (informations en entrée et informations en sortie).
- seuls les événements les plus courants seront présentés.
- il ne s'agit pas de présenter les règles ergonomiques de manière exhaustive.

○ Documents fournis

- documentation sur les principaux contrôles graphiques pour applications de bureau (applications fenêtrées)
- documentation sur les formulaires HTML

