

STUDY OF UML DIAGRAMS USING AN APPLICATION WHICH SIMULATED THE AUTOMATIC COFFEE MACHINE

POPA SORIN EUGEN^{1*}, PUIU PETRU GABRIEL¹

Universit  VasileAlecsandri  du Bac  u, CaleaM  r       156,Bac  u, 600115, Roumanie

Abstract: This work aims to achieve applications that simulate the operation of a coffee machine. This application is designed to allow students to learn logic information technology the use of UML diagrams in programming engineering discipline.

Keywords: UML, software engineering diagrams, C#

1. INTRODUCTION

Cet article pr  sente une application en C# sous Microsoft Visual Studio 2012, qui fa  onne le fonctionnement d'une machine    caf   automatique en utilisant les principes de programmation orient  e objet et de g  nie logiciel. D  veloppement d'applications s'effectue en respectant les techniques apprises de programmation et g  nie jouent un r  le important dans le processus enseignement/apprentissage contexte ci-dessus. Le fonctionnement de l'application et la mani  re dont, les   tudiants doivent comprendre la fa  on dont les diagrammes UML. Ci-dessous, nous pr  sentons bri  vement quelques aspects fondamentaux de certains des diagrammes UML.

N'oubliez pas que le graphique utilise des cas d'utilisation [8] :

- pour mod  liser le contexte d'un syst  me : d  termination des limites du syst  me et des acteurs avec lesquels elle interagit.
- pour mod  liser une configuration requise : ce que vous devez faire syst  me (d'un point de vue en dehors du syst  me) qui est ind  pendant de la fa  on dont il est cens   pour faire. Pour sp  cifier le comportement entra  nera. Le syst  me appara  t comme une bo  te noire. Ce qui est vu, c'est comment il r  agit aux actions de l'ext  rieur.

Diagramme de classes est utilis   pour mod  liser la structure (statique sur la vision) d'un syst  me. L'organigramme contient les classes et interfaces, les objets et les relations qui sont   tablissent entre eux. Relations peuvent   tre de type : Association, agr  gation, g  n  ralisation, d  pendance, r  alisation.

Classes sont utilis  es pour capturer le syst  me pour   tre d  velopp  s de vocabulaire. Ils peuvent inclure :

- abstractions qui font partie du probl  me ;
- n  cessaire    l'  poque des classes de mise en   uvre.

Diagramme de l'  tat est utilis   pour mod  liser le comportement d'un objet unique. Tableau des   tats sp  cifie qu'une s  quence d'  tats un objet traverse au cours de sa vie en r  ponse    des   v  nements, ainsi que la r  ponse    ces   v  nements.

2. MONTAGE EXP  RIMENTAL

Le sch  ma logique de fonctionnement de la machine    caf  , illustr  e    la Figure 1, est une application permettant de simuler le fonctionnement de la machine    caf  . Le panneau avant de l'application est illustr      la Figure 2.

* Corresponding author e-mail: sorin.popa@ub.ro

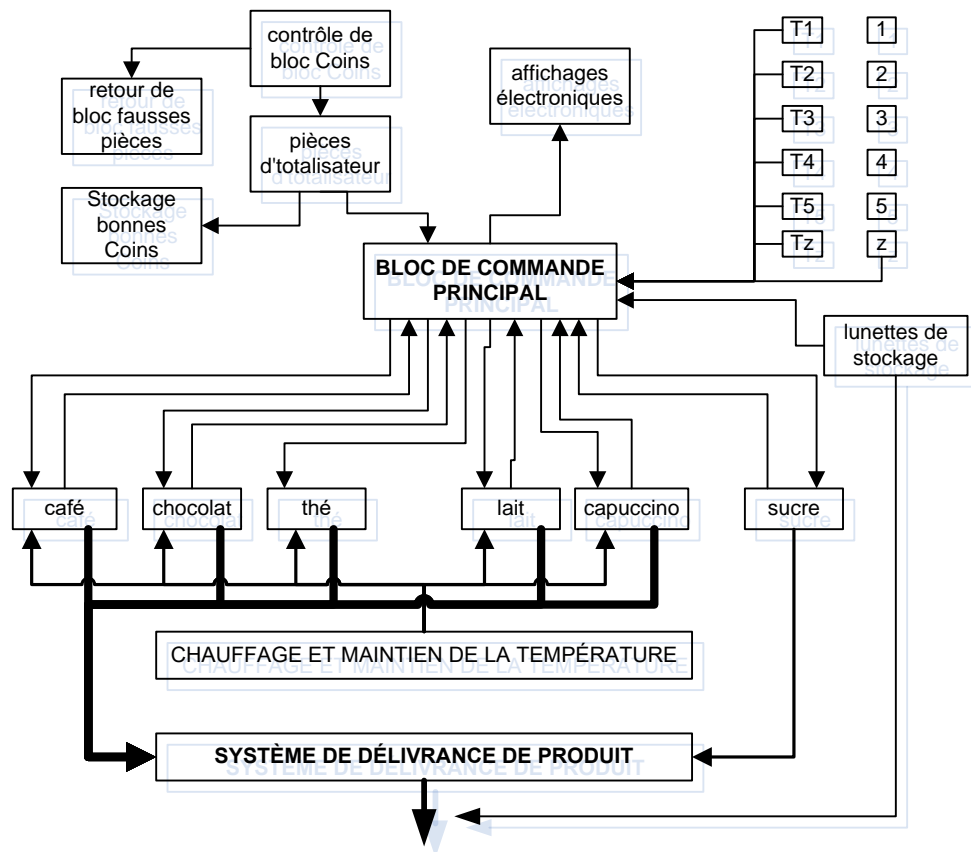


Fig.1. Le schéma logique de la machine à café.

Fig.2. La façade de l'application automatique de café.

Mode de fonctionnement du programme est le suivant : Le formulaire principal est chargé, il appelle la fonction `CheckForEnoughMoney()` qui vérifie si la boisson est supérieure au montant inscrit dans le dispositif et active les touches correspondantes sur les produits dont le prix est inférieur au montant que vous avez entré.

Fonction CheckForEnoughMoney() est une cascade d'if-else qui vérifie un coût en temps de chaque produit. Les prix des produits sont stockés dans un tableau « coût ».

Sur le panneau, il y a 4 boutons, chacun associé à une somme d'argent, qui simule les montants d'argent que nous pouvons entrer dans le dispositif. Lorsque vous appuyez sur un des boutons, la quantité augmente de la valeur associée à chaque bouton sur le panneau pour afficher le montant total saisi et appelle de nouveau la fonction CheckForEnoughMoney(), qui active les produits pour lesquels le montant inscrit est supérieure au prix du produit.

Le Comité estime en outre que simule un curseur pour ajuster le niveau de sucre et une progressBar montrant les progrès accomplis dans la préparation de l'appareil sélectionné. Sélection des produits se fait en appuyant sur le bouton. Sur simple pression d'un bouton pour la sélection d'un produit, il fait la différence entre le montant que vous entrez et le coût du produit en calculant le reste, appelez la fonction CheckForEnoughMoney () pour voir que si vous avez l'argent qui reste dans l'appareil photo pour un autre produit, le bouton Démarrer est activé et le progressBar "libère le reste" .

3. RÉSULTATS ET DISCUSSION

Sur la base de la demande de tracer les principaux diagrammes UML [6], [7]. La figure 3 illustre les cas d'utilisation de diagramme. Dans ce schéma, les deux acteurs sont présents, le fournisseur et le client. Le fournisseur de service a pour mission de soutenir la machine à café, à savoir, il assainit il nourrit avec différentes sortes de produits (café, lait, sucre, verres, etc.) et gère le monétaire (recueille de l'argent venant et insérez les pièces de monnaie pour le reste). L'utilisateur peut entrer dans l'argent, choisissez un produit et servir le produit.

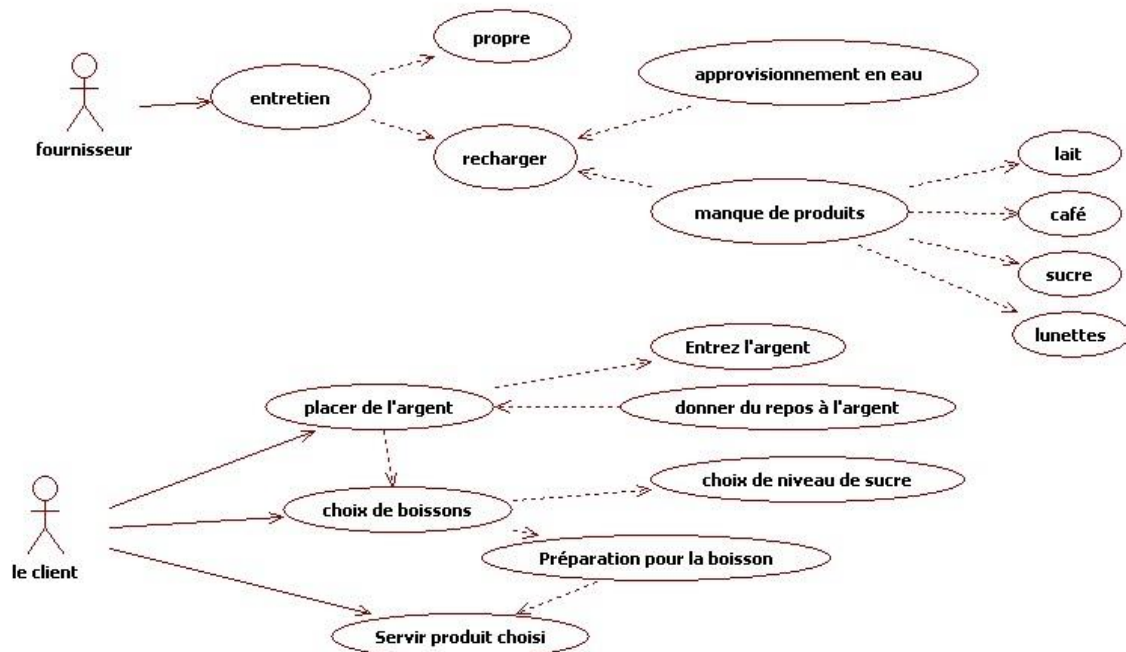


Fig. 3. Le diagramme des cas d'utilisation.

Dans la Figure 4 montre le graphique des activités du point de vue du client. Il introduit l'argent, mis en place le niveau de sucre, sélectionnez un produit, choisissez un produit, si l'argent n'est pas assez d'argent marketing est retourné, s'il n'y a qu'aucun produit sélectionné n'est à choisir un autre produit. Si le produit est là et l'argent n'est pas suffisant, la préparation du produit. En fin de compte, le client récupère la préparation du produit.

La figure 5 illustre le diagramme de classes, comme a été implémentée dans la demande.

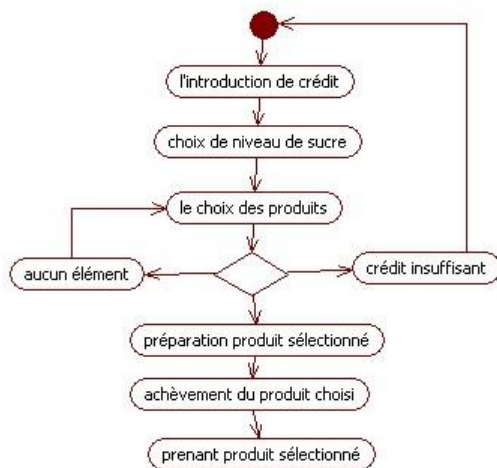


Fig. 4. Diagramme d'activité.

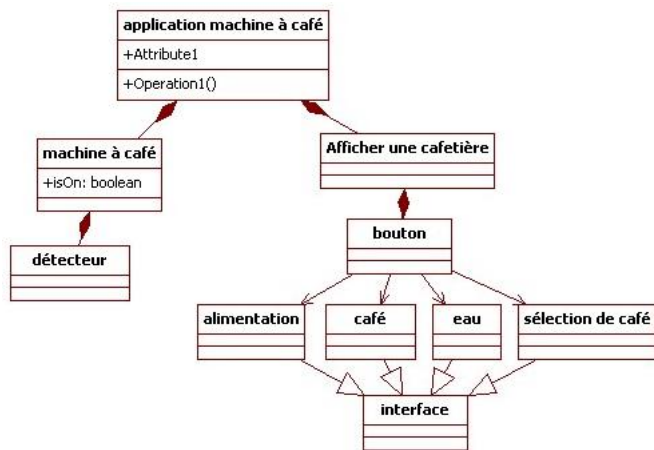


Fig. 5: Diagramme de classes.

4. CONCLUSIONS

En effectuant ce travail, il chercha un enseignement de méthodes peuvent être utilisées dans le processus d'apprentissage, d'utiliser des diagrammes UML. L'application est un outil pour les étudiants de troisième année de programmation préalablement au symposium enseignements technologie spécialisation discipline d'ingénierie.

Les méthodes utilisées dans l'application sont enseignées selon ce qui précède mentionné, et la demande est un exemple de mettre en pratique les notions théoriques.

Le programme peut être amélioré et développé, principalement par le biais de l'ajout de nouvelles fonctions, comme l'attribution d'argent placé à huis clos si j'ai changé d'avis sur l'achat d'un produit de la machine distributrice. À l'avenir, il veut une implémentation à l'aide de matériel et équipement, et ici nous envisageons la possibilité d'obtenir un formulaire en utilisant un Arduino avec plates-formes ou Raspberry Pi.

RÉFÉRENCES

- [1] Cornelia Novac Ududec, Ingineriasistemelor de programare :ingineriaprogramarii, Bacau: Editura Alma Mater, 2011, ISBN: 9786065271265,
- [2] McGregor J., D. Sykes – A Practical Guide to Testing Object-Oriented Software, Addison Wesley, 2001, 417 p., ISBN -0-201-32564-0.
- [3] Fowler M., – UML Distilled – A Brief Guide of the Standard Object Modeling Language, Addison Wesley, 2003, 208p, ISBN-0-321-19363-7
- [4] Booch G., J. Rumbaugh, I. Jacobson – Unified Modeling Language User Guide, Addison Wesley , 2005, 496p, ISBN- 0-321-26797-4.
- [6] Culea, G.; Bucur, I., Popescu, Tracking system for photovoltaic panels realized with Twido controller, Annals of DAAAM for 2007 & Proceedings, vol. 18, 201-202, NO. 1, PP 296, Editor Branko Katalinic, Published by DAAAM INTERNATIONAL, Vienna, Austria 2007, ISSN 1726-9679, ISBN 3-901509-58-5
- [7] Popescu, C.; Culea, G., Bucur, I, Wireless embedded application for automation systems, Annals of DAAAM for 2007 & Proceedings, vol. 18, 587-588, NO. 1, PP 296, Editor Branko Katalinic, Published by DAAAM INTERNATIONAL, Vienna, Austria 2007, ISSN 1726-9679, ISBN 3-901509-58-5
- [8] Culea George, Modelling based on synchronized Object-Oriented Petri net, Recent Advances in Automatic Control, Information and Communication, Recent Advances in Electrical Engineering, ISSN 1790-5117, 2013, series 19, pag 209-2014