

C# Partial Classes as a Mechanisms to Implement Feature-Oriented Designs: An Exploratory Study*

Lidia Fuentes
Dpto. Lenguajes y Ciencias de
la Computación
Universidad de Málaga
(Spain)
lff@lcc.uma.es

Elio López
Dpto. Lenguajes y Ciencias de
la Computación
Universidad de Málaga
(Spain)
ealopezs@gmail.com

Pablo Sánchez
Dpto. Matemáticas,
Estadística y Computación
Universidad de Cantabria
(Spain)
p.sanchez@unican.es

ABSTRACT

C# partial classes allows developers to divide the implementation of a class into several slices where each slice contains an increment of functionality as compared to the other slices. Thus, combining different set of slices, we can get classes with a variable range of functionality. With this description, C# partial classes seems to be, as also pointed out by other authors, a suitable mechanism for implementing feature-oriented designs. This paper explores this idea, by systematically applying C# to a feature-orientate decomposition based on an industrial case study and comparing the results we previously obtained using the feature-oriented language CaesarJ. As main contributions, (1) we identify benefits and pitfalls of C# partial classes for implementing feature-oriented decompositions; and (2) we outline potential solutions to alleviate these pitfalls.

Categories and Subject Descriptors

D2.2 [Design Tools and Techniques]

General Terms

Design, Languages

Keywords

Partial Classes, Feature-Oriented Programming, Software-Product Line

1. INTRODUCTION

Pues eso, introduccion.

*This work has been supported by the Spanish Ministry Project TIN2008-01942/TIN, the EC STREP Project AM-PLI IST-033710, and the Junta de Andalucía regional project FarmWare TIC-5231.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

FOSD'10, October 13, 2010, Eindhoven (The Netherlands)
Copyright 2010 ACM ...\$5.00.

2. WHAT WE WANT TO FIND IN A FEATURE-ORIENTED PROGRAMMING LANGUAGE

Básicamente, características que tienen los lenguajes como CaesarJ que nos facilitan la vida para implementar modelos de SPL, tales como lo de reescritura automática de las dependencias entre clases.

3. CASE STUDY: A SMART HOME SOFTWARE PRODUCT LINE

Copia y pega de la descripción de la Smart Home más breve explicación de los modelos arquitectónicos. La figura del modelo hay que ponerla después de escribir el artículo, para que contenga todos los elementos que nos hacen falta para explicar los fallos de C#.

4. IMPLEMENTING THE SMART HOME MODELS USING C# PARTIAL CLASSES

Describir como se implementan diferentes situaciones en el modelo de la SmartHome.

4.1 Scenarios 1: ...

Descripción de como implementar algo tal como una feature que extiende de otra.

4.2 Scenarios 2: ...

Descripción de como implementar otra cosa, tal como una feature que extiende de dos.

4.3 Scenarios 3: ...

Otro caso diferente ...

4.4 Conclusions

Tabla o bullets comparando lo que queríamos tener y lo que tenemos con las clases parciales.

5. SKETCH OF SOLUTIONS FOR C# PARTIAL CLASSES PITFALLS

Breves comentarios sobre como solucionar estos problemas. (NOTA: No estoy muy seguro acerca de si poner o no esta sección).

6. SUMMARY AND FUTURE WORK