

ABSTRACT:

UNDERSTANDING DESIGN PATTERNS AS NORMAL FORMS

Gábor Kusper, gkusper@aries.ektf.hu
Eszterházy Károly College

Design Patterns, DP-s for short, describe a recipe for the programmer how to develop an efficient, reusable and easy to extend program, like a pottage recipe tells you how to make a delicious pottage soup. The DP-s got interest after coming out the book of the gang of four (*Design Patterns, Elements of Reusable Object-Oriented Software*) in the 90'. In the meantime they achieved their rightful place in the practice and also in the education. As the same time it is a problem how to formalize them. The main reason of this, that there is no commonly agreed model which is even good for education purposes.

The proposed models try to formalize the basic concepts of Object Oriented Programming, OOP for short, because this is the language of DP-s. The descriptions of DP-s in these models are awkward.

We propose a new model which gives a correspondence between Relational Data Model and OOP as follows: Table / Class, Entity / Object, Foreign Key / Reference.

This is appropriate, because the Rational Data Model is well studied and is suitable for education, because it is easy to explain its concepts by real world examples.

For example the Normal Forms are easy to understand as follows: Their task is to reduce dependences, to allow for example insert without many changes in the model.

It is alike easy to understand DP-s from this view point: Their task is to reduce dependences, to allow for example program extension without many changes in the code.

One can see that there is parallelism between DP-s and Normal Forms, which can be shown formally in the proposed model for example in case of the Flyweight design pattern.