Семантика на проблемно-ориентирания Model-View-Controlling

Петър Сигалов, Адриан Митев

Semantics of Problem-oriented Model-View-Controlling: *MVC is a design pattern which has originated from the Smalltalk (in person by Trygve Reenskaug). If this pattern: A model is an object representing data or even activity; A view is some form of visualisation of the model's state; A controller offers facilities to change the state of the model. The controller interprets the mouse and keyboard inputs by the user's side, commanding the model and/or the view to change as appropriate. In Java technology MVC is applied as Java-Swing ideology and further in a wider scope in JSF Framework. In this article, the MVC pattern is used as an approach to solve tasks in the field of CAD-area.*

Key words: *MVC, Java-Swing, Java-Enterprise, JavaServer Faces, CAD-Approaches, CAD-Methodic*

ВЪВЕДЕНИЕ

Парадигмата Model-View-Controlling (MVC) е въведена като подход на организация за първи път от Xerox във връзка с езика Smalltalk (един от първите обектно-ориентирани програмни езици) [1]. Основен стремеж при тази мострена структура е създаването на независимост между три фундаментални части при изграждането на софтуерно осигуряване в една приложна област, като тези части са назовани съответно: Модел, Презентация и Управление. Веднага трябва да се отбележи, че тази независимост предполага точно дефиниране на връзките между тези три софтуерни части. Концепцията на независимост на частите и на дефинирането на връзките трябва да даде възможност за паралелност при създаването и за лекота при разширяване и поддръжка на цялостната програмна разработка. Един съществен приност се проявява в скоростта и скалируемостта на приложенията разработени в духа на тази идеология.

MVC – база на съвременните технологии и Frameworks

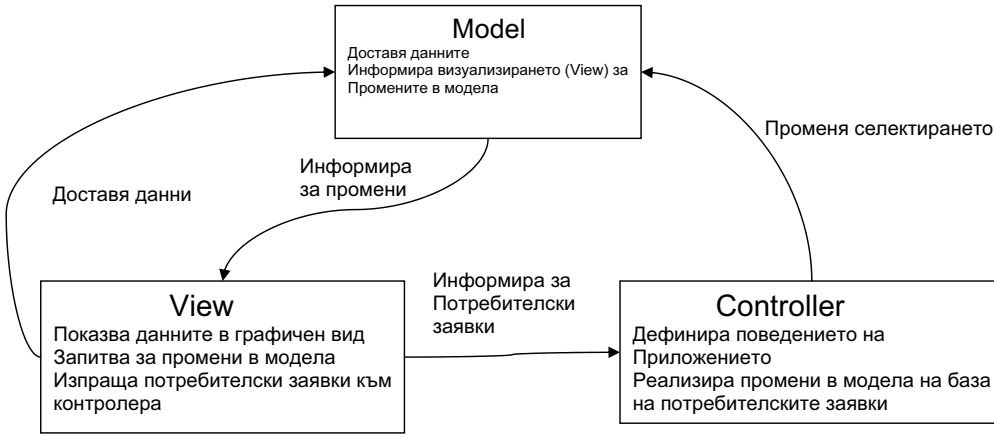
Smalltalk се явява като един фаворизиран програмен език, до неговото практическо изместване от Java, който вече, като език и като технология, притежава значително по-мощни средства – както по отношение на широтата на областите на програмните разработки, така и по отношение на възможностите за елегантна имплементация по време на дизайна.

1. MVC и Java-Swing

В Java, подходът MVC се явява като градивен принцип най-напред в своя най-опростен вид в GUI-програмирането - в библиотеката на Swing-класовете. При относително по-простите от тези класове неговата проява е например в методите getDocument и setDocument, с които документираността (моделът) постигната в инстанцията на един клас, може да бъде предадена на инстанция от друг клас и впоследствие да бъде обработвана и визуалзирана от средствата на този клас. При най-комплексните от от Swing-класовете обаче – като JTable и JTree, предаването на модела води не само до по-добро програмно структуриране и до повишаване на скоростта, а и до едно съществено разнообразие на възможностите за визуализация, което – в разнообразните приложни области, може да се реализира като много плътна адекватност с проблемно-ориентираното мислене на потребителя. Фигурата 1 по-долу, дава представа за принципа на MVC и за неговата реализация в Java-Swing.

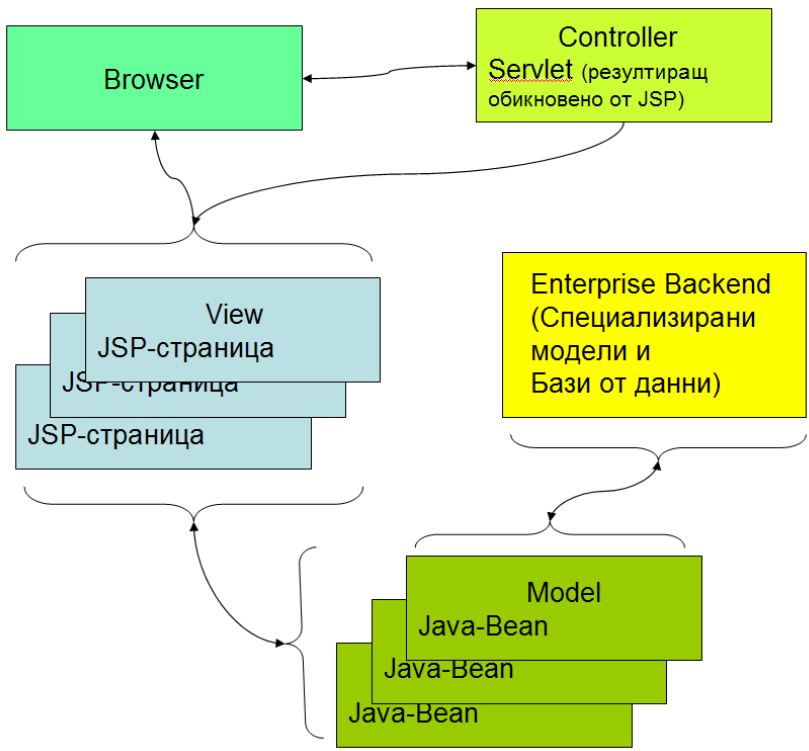
2. MVC в трислойните интернет-архитектури

Понастоящем, а като детайлен принцип най-вероятно в доста дългосрочен план, за най-разпространена трислойна архитектура можем да разглеждаме Framework-а JavaServer Faces. Удачно е да отдадем централно място на приложни-



Фиг. 1 Идеята за Model-View-Controlling като база за конкретни реализации в GUI-програмирането и в съвременните Frameworks

те модели – представлявани от Java-Beans, които се намират в средния слой – application-сървъра и които в своята работа ползват модели и данни и от т.нар. отдалечен сървър. Конт-

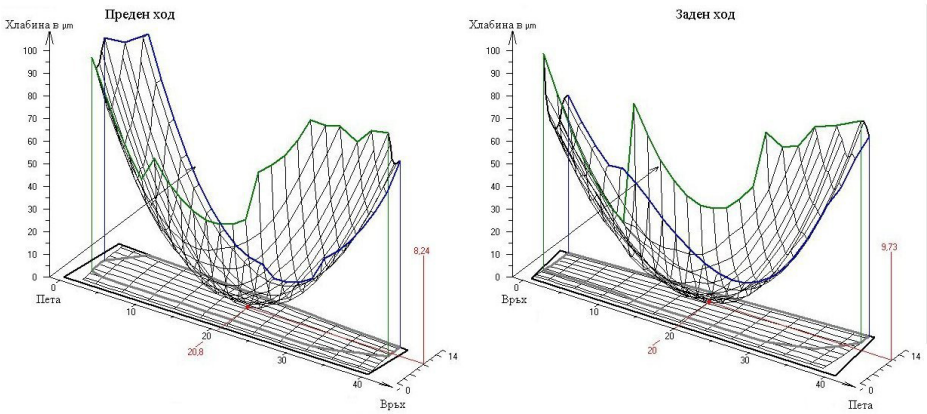


Фиг. 2 Принципна реализация на MVC в трислойната архитектура на Java-Enterprise

ролерът принципно се представлява от сървлет/и резултиращ/и от една или повече JavaServer Page/s. Той комуницира с потребителския слой ползвайки изображения разработени посредством JavaServer Page/s, като в зависимост от логиката на решаваната задача или от потребителската заявка избира съответната JSP и начина на нейното конкретно представяне.

Семантично надграждане на парадигмата MVC в една приложна област

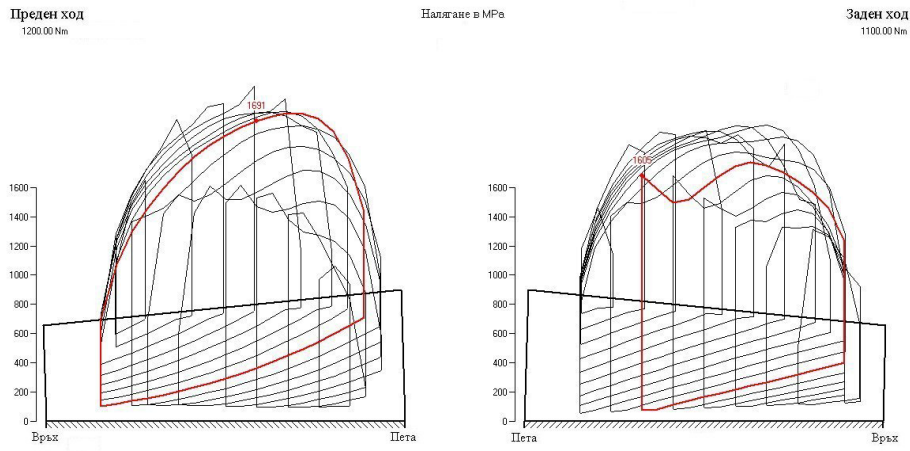
Удачно е да се разглежда една конкретна и голяма приложна област, като се правят разсъждения върху една нейна базова разработка, изискваща разнообразни графични представяния, за подпомагане на решенията, вземани от специалистът който ще я използва. Като база е ползвана разработка [4] за конструиране на хипоидни предавки, която е поръчана на TU-Dresden/Германия – институт по конструиране и CAD от концерна Hurth-modul GmbH, който е производител на зъбонарезни машини с CPU. Целта на нашите разглеждания е MVC-конципирането



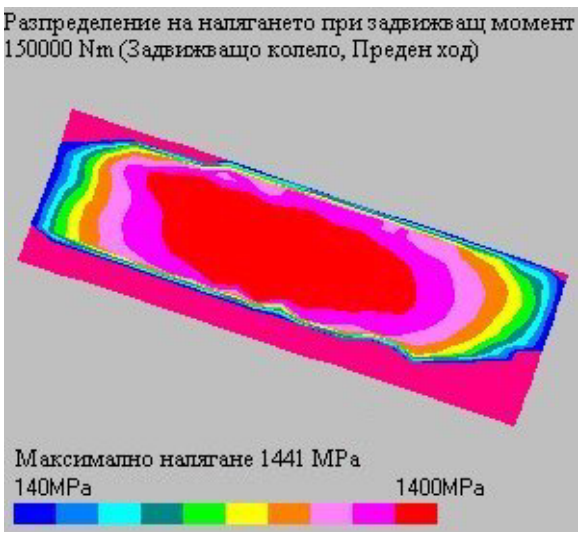
Фиг. 3 Извеждане на данни за разпол. на контактната точка при хипоидна предавка

Фиг. 4 Данни за разположение на петното на контакта при хипоидна предавка

Фиг. 5 Данни за възможната неравномерност на хода при хипоидна предавка



Фиг. 6 Данни за повърхностното налягане при хипоидна предавка



Фиг. 7 Изометрично представени данни за повърхн. налягане при хипоидна предавка

на програмната разработка. Това надграждане, което е в проблемно-ориентираната област, трябва да е адекватно на базовата софтуерна MVC-идеология. То изисква нейното разработване в конкретната CAD-реализация, което има своя специфика, изграждане съвместно със специалистите от страната на възложителя. Съгласно тази специфика, която впрочем има значителна степен на универсалност в областта на CAD-разработките, контролингът не може да бъде изцяло предварително програмиран. Една част от него преминава в областта на човеко-машинния интерфейс, като това се запазва, въпреки имплементирането в последно време на елементи от Изкуствения интелект в програмираната част на контролинга. Контролирането на процеса на конструиране от страна на потребителя се отразява в програмното осъществяване на изгледите (view-частта). По принцип формират две основни групи: Информационни изображения – които служат за подпомагане на

решенията на потребителя (изведената в човеко-машинен интерфейс част от controlling-а), и такива представящи окончателната графична и числена информация. Докато последната е обичайно известна и подчинена на приложни стандарти в областта на потребителя, то тази, която подкрепя неговите разсъждения по време на процеса на конструиране (при ползване на програмната разработка) обикновено е нова, изгражда се при съвместни консултации между двете групи от специалисти и представлява съществения приност на идеологията на MVC в нивото на CAD-разработките. Изображенията от 3 до 7 са именно такива. Те са получени при работата на програмния продукт упоменат в [4] и представят едно информационно ниво от решаване на конкретна задача с последния. Този "View" е необходим за творческия на "Controlling-а" от страна на потребителя.

ЗАКЛЮЧЕНИЕ

Осъществяването на MVC в по-високите нива – тези на проблемната ориентираност, е предпоставка за ефективността на тази идеология на базово равнище, а с това и за нейната ефективност в конкретната CAD-разработка като цяло. Разработването на MVC в проблемно-ориентираната област изисква съвместна работа със специалисти от областта, които, образно казано, знаят какви са информационните необходимости на решаваната задача и посредством работата си с програмните реализатори научават какви са възможностите за изграждане на съответните диалогови информационни картини (проблемно-ориентиран "View"). Оттам се извеждат възможностите за решения в диалогов режим (съществена част на проблемно-ориентиран "Controlling"). Извеждането на общовалидни принципи е обект м/у другото и на дисциплината „Конструкционна информатика". По такъв начин се осъществява плавното преминаване на идеологията на MVC през различните нива на проектантската дейност, като: семантиката на проблемно-ориентираната област и полагането на проектните модули на програмната разработка, планирането на програмната архитектура и, в резултиращия етап - осъществяването на софтуерната имплементация. Това помага за изграждането и използването в приложните области на подходи и алгоритми – насочени към програмно реализиране, както и за влагането на преизползваеми компоненти в програмната имплементация.

ЛИТЕРАТУРА

[1] Trygve M. H.: Reenskaug: http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html. Последно ползване: м. март 2009)

[2] http://www.dpunkt.de/java/Programmieren_mit_Java/Oberflaechenprogrammierung/40.html, Последно ползване: м. август 2010

[3] Lahres, B., Rayman G.: Praxisbuch Objektorientierung – Professionelle Entwurfsverfahren, Galileo Computing, ISBN 3-89842-624-6

[4] Linke, H., Schlecht, B.: TU-Dresden, IMM, Jahresbericht 2001, Softwareentwicklung fuer Firma "Hurth Modul GmbH"

За контакти:

доц. д-р Петър Сигалов, Русенски университет "Ангел Кънчев", кат. Информатика и информационни технологии, тел.: (082) 888754, e-mail: sigalov@ami.uni-ruse.bg

маг.-инж. Адриан Митев, Сирма Ай Ти Ти, София бул. Цариградско шосе 135, email: adrian.mitev@sirma.bg

Докладът е рецензиран.