|  |  |
| --- | --- |
|  | ВЕЛИКОТЪРНОВСКИ УНИВЕРСИТЕТ  “СВ. СВ. КИРИЛ И МЕТОДИЙ” |

Владимир Йорданов Йотов

Модели, базирани на йерархични композиции от пространства, за управление на софтуерни версии

Автореферат

на ДИСЕРТАЦИЯ

за присъждане на образователна и научна степен “Доктор”

по научна специалност: 01.01.12 “Информатика”

Научен ръководител: доц. д-р Хр. Тужаров

Велико Търново  
2012

# Въведение

## Актуалност

Управлението на версията на софтуерните продукти заема важно място в областта на софтуерното инжинерство [G365, G100]. Въпреки наличието на разработени модели, научно-приложната област предоставя възможности за търсене на решения за постигане на по-висока ефективност на работният процес. Модерните гъвкави методологии предлагат един по-свободен начин на развитие на софтуерните продукти, като те предполагат използването на специалисти от високо ниво, познаващи разработваният продукт и използваната технология в детайли. Създаването и задържането на екипи от такива специалисти представлява предизвикателство за всяко едно корпоративно ръководството. Настоящата дисертация може да се разглежда като опит да се предостави възможност за снижаване на рисковете от използването на такива кадри и въвеждането на инструментариум за автоматизация при използването на гъвкавите методологии.

## Обект и методология на изследването

Обект на изследване на дисертацията са моделите и методите в управлението на версии чрез използването на йерархично композирани работни пространства за постигане на по-ефективен подход на нейното управление, ускоряване анализа на влиянието на промените над системата, усъвършенстване политиката на управление на знания в компаниите, и инструмент за обсъждане финансовите аспекти на проектите.

Методология на изследването включва следните подходи: анализ на целите и задачите, стоящи пред моделите в научно-приложната област; сравнителен анализ на използваните модели и методи и определяне на нови идеи; търсене, изследване и развитие на модели и методи за управление на версия и повишаване ефективността на процеса на създаване и поддържане на софтуерните продукти.

## Цели и задачи дисертацията

**Цел:** Да се изследват, създадат и развият на модели за управление на софтуерни версии в среда, базирана на йерархично композирани на работни пространства.

Във връзка с основните цели се поставят следните задачи:

1. Да се създаде модел на версионизиран обект, осигуряващ пълна гъвкавост при дефиниране степента на гранулираност на данните в съчетание с простота и универсалност.
2. Да се създаде модел на среда с йерархично композирани работни пространства, както и да се определят правилата за управление на версия на обекти в тази среда.
3. Да се адаптира на метод за проследимост базиран на събития за модела на среда с йерархично композирани работни пространства.
4. Да се определи на терминологията в областта на версионизирането с използването на йерархично композирани работни пространствал
5. Да се създаде методологична рамка за създаване на софтуерни продукти в среда с йерархично композирани работни пространства.
6. Да се докаже повишаване на ефективността при разработването на софтуерни продукти при разработените модели.

## Структура на дисертационния труд

Дисертацията се състои от увод, три глави, заключение, използвана литература, две приложения и прототип.

**В първа глава** е направен обзор на моделите в областта на управлението на версиите. Разгледани са място, целите и задачите на управлението на версии в рамките на разработването и поддържането на софтуерни продукти. Направен е обзор на съществуващите модели на версионизираните обекти, и начинът на тяхното съхратняване в репозиторито с версии. Отелно е направен обзор на темата за съвместната работа на сътрудниците, където е наблегнато на работните пространства, като средство за осъществяването на кооперираност. За постигане пълнота на обзора са сравнени методите за проследимост на промените. Главата завършва с определяне на изводите, формиране на целта и задачите на дисертацията, които следва да бъдат решени във втора и трета глава.

**Във втора глава** са представени теоритичните модели за управление на версия в среда с йерархично композирани работни пространства. Моделите са допълнени с авторска методологична рамка за тяхното ефективно използване. Във формулираните в края на главата изводи са посочени предимствата на разработените модели.

**Трета глава** съдържа аналитичен избор на средства за реализиране на програмен прототип на система, реализираща теоритичните модели. Представени са описания на авторска алгоритмичната реализация на по-важните моменти от прототипа. В главата е направена е експериментална сравнителна симулация на разработка на програмен продукт със и без използване на разработеният прототип при гъвкава методология за разработване. Във формулираните в края на главата изводи са посочени предимствата на разработените модели.

В заключението са направено обобщение на получените резултати. Формулирани са основните резултати в рамката на дисертацията. Посочени са някои актуални задачи, които могат да бъдат естествено продължение на настоящето изследване.

Разработката и апробацията на резултатите са извършени самостоятелно, като регулярно са представяни в катедра „Компютърни технологии” на Великотърновски университет „Св. Св. Кирил и Методий”.

# Публикации, свързани с дисертационния труд

1. Jotov, V. An investigation on the approaches for version control systems. In Proceedings of the 9th international Conference on Computer Systems and Technologies and Workshop For PhD Students in Computing (Gabrovo, Bulgaria, June 12 - 13, 2008). B. Rachev and A. Smrikarov, Eds. CompSysTech '08, vol. 374. ACM, New York, NY, V.11-1. DOI= http://doi.acm.org/10.1145/1500879.1500959
2. Jotov, Vl., Transaction over Versioned Objects in Hierarchical Workspace Environment, ECAI09, 2009, Pitesti, Romania
3. Jotov, Vl., Towards a model of versioning domain, 5th Central and Eastern European Software Engineering Conference in Russia, Moscow, 28-29 Oct. 2009, ISBN 978-1-4244-5665-9, http://dx.doi.org/10.1109/CEE-SECR.2009.5501147
4. Йотов, Вл., Модел на данните в система за контрол на версии, базирана на йерархични работни пространства, 6 – 7. 11. 2009г, Велико Търново
5. Jotov, Vl., Adaptation of Event-Based Traceability Method for Environment with Hierarchal Composed Workspaces, SAI 2010
6. ~~Todorova,M., Todorov, G., Jotov,Vl., Principles of Open Sourse Software, In: Multinational Undergraduate Team Work. Exellence in International Capstone Projects. Edited by Nuno Filipe Escudeiro. Instituto Superior de Engenharia do Politecnico do Porto (ISEP/IPP) and Paula Maria Escudeiro. Instituto Superior de Engenharia do Politecnico do Porto (ISEP/IPP). IOS. Press. Amsterdam. Berlin. Tokyo.Washington, DC. 2011, pp.72-91.~~

# Съдържание на дисертационния труд

## Глава 1. Управление на версията при създаването на софтуерни системи

В първа глава е са разгледани съществуващите модели в областта на управлението на софтуерни версии. В резултат направеният обзор в се налагат следните изводи:

1. Системите за управление и контрол на версии представляват задължителен инфраструктурен инструмент в съвременното софтуерно производство. Промени в моделите на управлениета на версиите играе водеща роля над целият процес на създаване на софтуерни продукти.
2. Направен е анализ на различни модели за представяне и съхраняване на версионизирани обекти. Изтъкнати са предимствата и недостатъците на разгледаните модели. Определена е необходимостта от поддръжката на различно ниво на гранулираност при версионизираните обекти, обусловена от практиката на използване на няколко нива на абстракция при разработката на софтуерни системи. Нива на абстракция успешно може да се съпоставят на нивата на гранулираност.
3. Направен е анализ на различни подходи за съхраняване на версионизирани обекти. Подходът за съхраняване състояния на версионизираните обекти, предполага проста реализация и по-висока скорост системата.
4. Анализирани са предизвикателствата стоящи пред съвместната работа над един продукт. Определена е необходимостта от научно изследване на моделите и механизмите, използвани в областта на йерархично композираните работни пространства. Установено е, че йерархично композираните работни пространства служат като инструмент за осигурявана на автономна работа, като съчетават това възможност за коопериране на работата между участниците в процеса по създаване на софтуерни продукти.
5. Направен е анализ на темата за проследимост на промените. Представени са различните видове проследяващи връзки, както и методите за получаването им.

## Глава 2. Модели за управление на версии в среда с йерархична композиция на работни пространства

### 2.1. Модел на версионизиран обект

Водещите автори в областта на управлението и контрола на версиите [G42, G8] определят версионизираните обекти, като съставен от две части – състояния на обекта (версии) и граф на версиите, където Под граф на версиите се разбира такъв граф, чиито върховете представляват отделните състояния (версии) на обекта, а ребрата съответстват на логическата последователност на създаване на версиите.

Първата нова характеристика, която следва да притежава един модел на версионизиран обект е той да предоставя възможност да се определи нивото на детайлизираност, т.е. на гранулираност, която самият модел следва да поддържа.

1. Съставен обект ще наричаме обект, който е съставен от други обекти (версии на обекти) посредством композиции.
2. Композиция представлява същност определяща връзката между супер-обект и под-обект. Един съставен обект може да бъде супер-обект на една или повече композиции, т.е. да е съставен от един или повече под-обекти.



Фиг. Пример за промяни на съставността на обектите

Същността **версионизиран обект** е необходимо да притежава само и единствено уникален и непроменяем номер, който е удачно да се използва и като първичен ключ за същността. Версиите на един обект може да се разглеждат като негови примитиви (**версионизирани примитиви)**, чиито основни атрибути са следните:

* Глобален номер на версията
* Номер на версионизиран обект, с който дадената версия е свързан;
* Номер на версия – пореден номер, който определя по уникален начин версията в рамките на обекта.
* Наименование на обекта. Определяйки наименованието на ниво примитив, потребителят получава възможност да проследява отделните версии даже и при преименоване на обектите. Така полученият модел става по-пълноценен, елиминерайки недостатъка свързан с преименоването на обектите (файловете) при системи като CVS, SVN, Git, Metcury и др. [G50, G54, G101, G387, G389].
* Съдържание на обекта, включващи данните в съответната версията на обекта.

Версионизираният примитив се определя еднозначно посредством уникалната двойка **номер на версионизиран обект** и **номер на версия.**

За нуждите на версионизиране на съставни обектите, следва да се дефинира допълнителна същност - „Композиция на версионизирани примитиви” (накратко композиция), която е еднозначно свързва версията на супер обекта с версиите на неговите под-обекти. Атрибутите на същността са:

* Глобален номер на супер-обекта;
* Глобален номер на под-обекта

За нуждите на отчетността и проследимостта на промените, така създаденият модела следва да се разшири с цел да да поддържа граф на верссите. В ER моделите е прието графовата структура да се моделира от две същности – същност на възлите и същност на дъгите [G410].Тук дъгите на графа (показващи прехода от една верися в друга) следва да притежават следните атрибути:

* Номер на дъгата – първичен ключ за същността
* Глобален номер на изходната версия
* Глобален номер на целевата версия
* Потребител, извършил промяната
* Дата и час на промяната
* Допълнителни данни относно промяната



Фиг. ER модел на версионизиран обект

#### 2.1.1. Версионизиране на съставен версионизиран обект

Настоящата подсекция има за цел да представи особеностите при управлението на версия на съставни обекти от първи ред. Базирайки се на тях, ще се определи процесът на версионизиране на съставни обекти от ред N.

1. Съставен обект от ред **0** ще наричаме такъв обект, за който няма асоциирани под-обекти, т.е. фактически той не се явява съставен обект. Съставен обект от ред **N** ще наричаме такъв обект, за който най-големият ред на асоцииран под-обект е равен на **N-1**.
2. Един под-обект сам по себе си може да се явява съставен обект от други обекти, като по този начин да се създаде суперпозиция от съставни обекти.

Една от основните задачи, която стои пред настоящият научно-приложен труд е да не усложняваме без необходимост тук създадените модели. Изхождайки от това, както и на факта от липсваща практическа необходимост, при построяването на суперпозиция от съставни обекти следва да въведем следното ограничаващи правила (ограничения):

1. В дадена суперпозиция от съставни обекти, обект може да присъства най-много един път.
2. Един обект може да присъства най-много в една суперпозиция от обекти.

## Глава 3. Изследване приложимостта на моделите

...

## Заключение

...

## Използвана литература

...