

## ИЗИСКВАНИЯ КЪМ УПРАВЛЕНИЕТО НА НАНОТЕХНОЛОГИИ В ЕЛЕКТРОНИКАТА

**Мила Илиева<sup>1</sup>**

<sup>1</sup>Минно-геоложки университет "Св. Иван Рилски", 1700 София, e-mail milailieva@abv.bg

**РЕЗЮМЕ.** В тази статия се синтезират изискванията на участниците в управлението на нанотехнологии в електрониката в рамките на един университет. За целта се прилага теорията на множествата. Изискванията към управлението се класифицират от гледна точка на: студент (дипломант), асистент (докторант), доцент (професор) и администратор на базата данни. Съществуващите модели дефинират специфични функции по функционални области, а тук се моделират функции за управление от високо ниво. Функциите са представени като случаи на използване на информационната система.

### REQUIREMENTS TO ELECTRONICS NANOTECHNOLOGY MANAGEMENT

**Mila Ilieva<sup>1</sup>**

<sup>1</sup>Mining and Geology University "St. Iv. Rilski", 1700, Sofia, e-mail milailieva@abv.bg

**ABSTRACT.** In this paper are synthesized the requirements of actors in electronics nanotechnology management in an university. For this reason the theory of multitudes is applied. The management requirements are classified from the point of view of: student(graduate), assistant (PhD), professor and database manager. The current models define specific management functions in functional areas; here are modeled high level management functions. The functions are represented as information system use cases.

### Въведение

Нанотехнология е област от приложните науки, която покрива широк диапазон от теми, като основните са контрол над веществото и създаване на устройства, съизмерими с размера на молекулите. Наноматериалите притежават характерни химико-физични свойства. Учените използват тези свойства за създаване на нови устройства и технологии. За тази цел трябва да има информационна система, която предоставя достъп до най-новите разработки.

### Методология

Условията на работа на системата за управление на нанотехнологии в университет се определят от няколко вида действащи лица: студент (дипломант), асистент (докторант), доцент (професор) и администратор на базата данни (Ернандес, 2007; Date, 2000; Hector, 2002). Администраторът на базата данни управлява цялата система и достъпа до нея. Доцентът (професорът) управлява нанотехнологиите и има изисквания към администратора за предоставяне на ресурси – софтуер и хардуер. Асистентът (докторантът) има изисквания към доцента (професора) за предоставяне на нанотехнологии и възможности за управление. За да се използват възможностите на информационната система, асистентът (докторантът) трябва да има достъп до ресурсите. Студентът (дипломантът) има изисквания към асистента. Той също трябва да има достъп до информационната система. За всяко действащо лице трябва да се синтезират услуги за управление. Изискванията на

действащите лица към съответната услуга за управление се синтезират чрез теорията на множествата. На базата на изискванията на участниците се дефинират функции за управление, като се прилага унифициран език за моделиране – UML (Фаулър, 2004).

Диаграмите в UML са класифицирани в две категории:

- Диаграми на поведение
- Диаграми на структура

От диаграмите на поведение най-често се прилага диаграмата на случаи на използване. **Случай на използване** е техника за определяне на функционалните изисквания на една система. Случаите на употреба описват типичните взаимодействия между потребителите на системата и самата нея, предоставяйки описание на начина, по който тя се използва.

От диаграмите на структура най-често се прилага диаграмата на класове обекти.

Тук за синтезиране на функциите, предоставени на действащите лица, се използват диаграми на случаи на използване. Взаимодействията между случаите на използване са «include» (съдържа) и „extend” (разширява).

Функционалните области за управление на информационни системи са: инсталиране, предоставяне, конфигурация, защита, поддържане, технически характеристики, таксуване и потребителски контрол (Magedanz, 1994).

## Резултати

Множествата с изискванията на действащите лица се означават както следва:

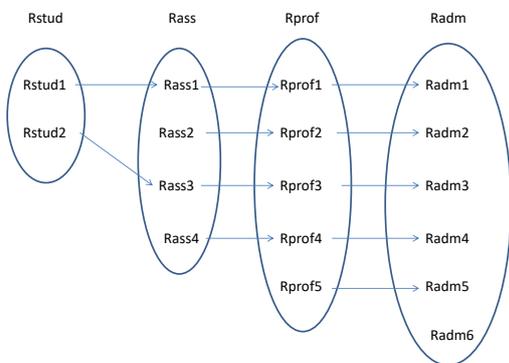
- R – множество с изисквания към цялата система;
- Radm – множество с изисквания на администратора на базата данни;
- Rprof – множество с изисквания на доцента (професора);
- Rass – множество с изисквания на асистента (докторанта);
- Rstud – множество с изисквания на студента (дипломанта).

Тогава според теорията на типовете може да се напише:

$$R = \text{Radm} \cup \text{Rprof} \cup \text{Rass} \cup \text{Rstud} \quad \text{и}$$

$$\text{Rstud} \subset \text{Rass} \subset \text{Rprof} \subset \text{Radm}$$

Между множествата с изисквания на участниците съществува инекция:



Фиг. 1: Инекция между елементите на множества Rstud, Rass, Rprof и Radm

Конкретните изисквания към управлението на информационната система са:

- Rstud – изисквания на студента (дипломанта) към управлението на системата:
  - Rstud1 – Изисквания за визуализация на управлението; Реализират се чрез функции за управление на профил на студент (дипломант);
  - Rstud2 – Изисквания за взаимодействие на функционалните области на системата; Реализират се чрез взаимодействие на функциите за управление на студент (дипломант);
- Rass – изисквания на асистента (докторанта) към управлението на системата:
  - Rass1 – Изисквания за визуализация на управлението; Реализират се чрез функции за управление на профил на асистент (докторант);
  - Rass2 – Изисквания за взаимодействия с подчинени участници – студенти (дипломанти);
  - Rass3 – Изисквания за взаимодействие на функционалните области на системата;

Реализират се чрез взаимодействие на функциите за управление на асистент (докторант);

- Rass4 – Изисквания за взаимодействия между ниво управление на нанотехнологии и ниво управление на ресурси; Реализират се чрез взаимодействие между функциите за управление на нанотехнологии, предоставени на асистент (докторант), и функциите за управление на ресурси, предоставени на администратора на базата данни;
- Rprof – изисквания на доцента (професора) към управлението на системата:
  - Rprof1 – Изисквания за визуализация на управлението; Реализират се чрез функции за управление на профил на доцент (професор);
  - Rprof2 – Изисквания за взаимодействия с подчинени участници – асистенти (докторанти); Реализират се чрез взаимодействие между функция за управление на профил на доцент (професор) и функция за управление на профил на асистент (докторант);
  - Rprof3 – Изисквания за взаимодействие на функционалните области на системата; Реализират се чрез взаимодействие на функциите за управление на доцент (професор);
  - Rprof4 – Изисквания за взаимодействие между ниво управление на нанотехнологии и ниво управление на ресурси (хардуер); Реализират се чрез взаимодействие между функциите за управление на нанотехнологии, предоставени на доцент (професор), и функциите за управление на ресурси, предоставени на администратора на базата данни;
  - Rprof5 – Изисквания за взаимодействия между ниво управление на нанотехнологии и ниво управление на определен ресурс (софтуер); Реализират се чрез функциите за управление на нанотехнологии, предоставени на доцент (професор), и функциите за управление на ресурси, предоставени на администратора на базата данни;
- Radm – Изисквания на администратора на базата данни към управлението на системата:
  - Radm1 – Изисквания за визуализация на управлението; Реализират се чрез функции за управление на профил на администратор на базата данни;
  - Radm2 – Изисквания за взаимодействия с подчинени участници – доцент (професор); Реализират се чрез взаимодействие между функция за управление на профил на администратор на база данни и функция за управление на профил на доцент (професор);
  - Radm3 – Изисквания за взаимодействие на функционалните области на системата; Реализират се чрез взаимодействие на функциите за управление на системата;

- Radm4 – Изисквания за взаимодействия между ниво управление на ниво хардуерни ресурси и ниво управление на софтуерни ресурси; Реализират се чрез взаимодействие между функциите за управление на хардуер за нанотехнологии и функциите за управление на хардуер изобщо;
- Radm5 – Изисквания за взаимодействия между ниво управление на хардуер и ниво хардуер; Реализират се чрез взаимодействие между функциите за управление на хардуер и функции, представящи определен хардуер;
- Radm6 – Изисквания за визуализация на управлението на хардуер за нанотехнологии; Реализира се чрез функция за конфигурация на хардуер за нанотехнологии.

## Дискусия

Синтезираните изисквания на участниците в управлението се трансформират във функции за управление. Всяка функция се представя като случай на използване, свързан с функционална област. Разработени са случаи на използване за функционални области Защита и Конфигурация. На **фигура 2** са показани случаи на използване за функционална област Конфигурация на нанотехнологии. Участниците са професор и асистент.

На професора са предоставени следните функции:  
 Функцията **ConfigNanotech(x)** служи за конфигуриране на нанотехнологиите.  
 Функцията **LoadNanotechComp(x)** служи за зареждане на компонент на нанотехнологиите в хардуера.  
 Функцията **LoadMaterials(x)** служи за зареждане на компонент Материали.  
 Функцията **LoadDesign(x)** служи за зареждане на компонент Технологии за производство.  
 Функцията **LoadDevices(x)** служи за зареждане на компонент Елементи.  
 Функцията **LoadMemory(x)** служи за зареждане на компонент Памети.  
 Функцията **LoadOptoelectronics(x)** служи за зареждане на компонент Оптиелектроника.

На асистента са предоставени следните функции:  
 Функцията **LoadNanotechData(x)** служи за зареждане на данни за нанотехнологиите.  
 На асистента са предоставени следните функции:  
 Функцията **ConfigAss(x)** служи за конфигуриране на асистента.  
 Функцията **InsertPIN(x)** служи за въвеждане на идентификационния код на асистента.  
 Функцията **InsertInterest(x)** служи за въвеждане на желана област от нанотехнологиите.

На **фигура 3** са показани случаи на използване за функционална област Защита на нанотехнологии. Участниците са професор и асистент.

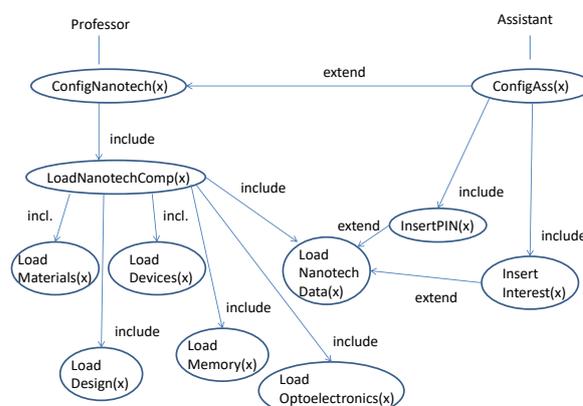
На професора са предоставени следните функции:  
 Функцията **SecureNanotech(x)** служи за защита на нанотехнологиите.  
 Функцията **InsertSecData(x)** служи за въвеждане на данни за защита на нанотехнологиите.  
 Функцията **InstallSecLog(x)** служи за инсталиране на журнал за защита на нанотехнологиите.

Функцията **WriteSecAlarm(x)** служи за записване на аларми за защита в журнала.

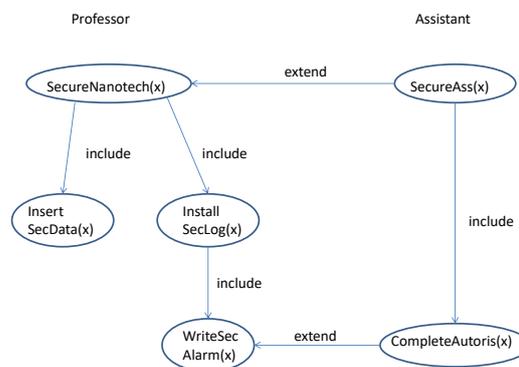
На асистента са предоставени следните функции:

Функцията **SecureAss(x)** служи за защита на асистента.

Функцията **CompleteAutoris(x)** служи за оторизиране на асистента.



Фиг.2: UML диаграма на случаи на използване на Конфигурация на нанотехнологии



Фиг. 3: UML диаграма на случаи на използване на Защита на нанотехнологии

## Изводи

В статията са синтезирани изисквания към управлението на нанотехнологии, съответстващи на отговорностите на всяко действащо лице: студент (дипломант), асистент (докторант), доцент (професор) и администратор на базата данни. Въз основа на изискванията към управлението са моделирани функции за управление от високо ниво. Въз основа на тези функции ще се моделират управлявани обекти с конкретни данни.

## Литература

- [1] Ернандес, Майкъл Х., "Проектиране на бази от данни", СофтПрес ООД, 2007
- [2] Фаулър, М., «UML. Основи», СофтПрес ООД, 2004г.
- [3] Date, J. C., "An Introduction to Database Systems", Addison Wesley Longman, Inc., 2000
- [4] Magedanz, T., "In integrated management model for intelligent networks", Munchen, Wien: Oldenburg, 1994
- [5] Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom, "Database Systems: The complete book", Prentice Hall, Inc., 2002