

УНИВЕРЗИТЕТ "СВ. КИРИЛ И МЕТОДИЈ"  
ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ - СКОПЈЕ

ЗОРАН Н. ЗДРАВЕВ

ОБЈЕКТНО ОРИЕНТИРАН МОДЕЛ НА  
ИНФОРМАЦИОНЕН СИСТЕМ ЗА  
ПРОЕКТИРАЊЕ, ГРАДБА И  
ЕКСПЛОАТАЦИЈА НА ТУНЕЛИ

\*\*\* МАГИСТЕРСКИ ТРУД \*\*\*

СКОПЈЕ, 1993

Ментор:

Данчо П. Давчев  
Електротехнички факултет  
Скопје

Членови на комисија:

Драган П. Михајлов  
Електротехнички факултет  
Скопје

Златко Д. Зографски  
Електротехнички факултет  
Скопје

Датум на одбрана: \_\_\_\_\_

Датум на промоција: \_\_\_\_\_

ЗОРАН Н. ЗДРАВЕВ

# ОБЈЕКТНО ОРИЕНТИРАН МОДЕЛ НА ИНФОРМАЦИОНЕН СИСТЕМ ЗА ПРОЕКТИРАЊЕ, ГРАДБА И ЕКСПЛОАТАЦИЈА НА ТУНЕЛИ

## АПСТРАКТ:

Во трудот е прикажан развојот на објектно ориентиран модел на информациона систем за проектирање, градба и експлоатација на тунели. Моделот е развиен со цел да се зголеми продуктивноста на инженерскиот кадар, како и да се искористи искуството од изградбата на други тунели. На почетокот се дадени краткиот преглед на постоечките системи кои се применуваат во тунелоградбата и основните поставки на објектно ориентираното програмирање. За развој на моделот е користена соодветна методологија. Тука е дефиниран проблемот, извршена е глобална анализа и се одредени главните подсистеми. Потоа за секој подсистем поединечно е утврден објектно ориентираниот модел. Креиран е дијаграм на токот на податоците, од каде се идентификувани објекти. Понатаму се утврдени атрибутите, видливоста и интерфејсите на објектите и на крајот е прикажан животниот циклус за секој објект. Моделот користи активни бази на податоци и работи на принципот на ЕСА (настан-услов-акција) правилата. Имплементиран е прототип на системот во C++, кој е тестиран на реален проблем.

## КЛУЧНИ ЗБОРОВИ:

Објектно-ориентирано програмирање; Тунели;

ZORAN N. ZDRAVEV

## OBJECT-ORIENTED MODEL OF AN INFORMATION SYSTEM FOR DESIGN, CONSTRUCTION AND EXPLOITATION OF TUNNELS

### ABSTRACT:

Developing of object-oriented model of an information system for design, construction and exploitation of tunnels is given in this article. This model was developed for increasing of productivity of engineers and using of experience obtained from other tunnels. The survey of existing information systems that are used in tunnel construction and general theoretical thesis of object-oriented programming is given in the beginning. The appropriate methodology was used in model developing. Problem definition and global analysis were done and the main subsystems were determined. Object-oriented model was determined for each subsystem. Data flow diagram was created and objects were identified. After that attributes, visibility and interfaces of objects were established. At the end, life cycle for each object was shown. The model uses active database with ECA (event-condition-action) rules. The system was implemented in C++ and tested on the real problem.

### KEY WORDS:

Object-oriented programing; Tunnels;

*Чувствувам должност на ова место да изразам благодарност кон сите оние кои ми помогнаа, посредно или непосредно, при изработката на овој труд.*

*Посебно би сакал да му се заблагодарам на мојот ментор проф. др. Данчо Давчев за сите совети околу пристапот, начинот на работа, како и конечно обликување на трудот.*

*Исто така посебна благодарност изразувам и кон проф. Милка Зафирова која несебично ми го пренесува своето богато искуство околу проектирањето и градењето на тунелите.*

*На крај би сакал уште еднаш да се заблагодарам на моето семејство за искажаната доверба и приемието за време на изработката на овој труд.*

*Скопје, 22.11.1993*

*Зоран Здравев*

## СОДРЖИНА

1. Увод.....	1
2. Преглед на постоечки информациона системи за примена во тунелоградбата.....	3
3. Елементи на објектно ориентиранот пристап.....	7
3.1. Основни концепти на објектно ориентираното програмирање .....	7
3.1.1. Апстракција.....	7
3.1.2. Енкапсулација .....	9
3.1.3. Хиерархија на класи .....	10
3.2. Развој на објектно ориентиран програм.....	13
4. Модел на информациона систем.....	15
4.1. Методологија за развој на информациона систем за проектирање, градба и експлоатација на тунели.....	15
4.2. Дефинирање на проблемот .....	18
4.3. Анализа на системот со глобален модел .....	26
4.4. Информационен подсистем "Проектирање" .....	28
4.4.1. Анализа на подсистемот.....	28
4.4.2. Објектно ориентиран модел на ИПС "Проектирање" .....	30
4.5. Информационен подсистем "Градба" .....	37
4.5.1. Анализа на подсистемот.....	37
4.5.2. Објектно ориентиран модел на ИПС "Градба" .....	40
4.6. Информационен подсистем "Експлоатација" .....	44
4.6.1. Анализа на подсистемот.....	44
4.6.2. Објектно ориентиран модел на ИПС "Експлоатација" .....	45
4.7. Подсистем за управување со активната база на податоци .....	47
5. Имплементација на системот.....	51
6. Заклучок .....	55

Литература

Прилог



# 1. УВОД

При проектирањето на било каков градежен објект, инженерите се соочени со обемни пресметувања. Поради специфичноста на тунелите, тие пресметувања можат да продолжат и во фазата на градба. Тоа се должи на фактот што во процесот на проектирање се влегува со многу непознати за видот и квалитетот на карпите низ кои се пробива тунелот.

Освен тоа, при проектирањето и при пробивањето на тунелите, многу повеќе се потпираме на емпириски методи отколку на пресметувања по строги физички закони. На сето ова се надоврзува и барањето за економична градба, поради што во последните неколку десетина години се развиени методи за пробивање на тунелите со активно следење на состојбата во карпите.

Поради сите овие факти, а за да се зголеми продуктивноста и економичноста на тунелоградбата, се јавува потреба од развој на информациона систем за проектирање, градба и експлоатација на тунели.

Како главни задачи на овој информациона систем се јавуваат:

- проектирање на тунели, со прогнозни тунелски притисоци и типови на подградба;
- следење на состојбата во ридскиот масив и споредување на реалната со проектираната геологија;
- следење на напрегањата и однесувањето на ридскиот масив со тек на време и
- формирање база на податоци, односно база на знаење, за подоцнежнo користење (како искуство) при проектирањето и изградбата на нови тунели.

Од друга страна, во последниве десетина години, комјутерската техника и информатиката навлегуваат во сите сегменти на животот и доживуваат масовна примена. Тоа се должи на динамичниот развој на хардверот како и на развојот на новите софтверски технологии.



Најактуелен пристап за развој на информациона системи, во последно време, е објектно ориентираното програмирање. Објектно ориентираното програмирање не претставува нова техника на програмирањето, туку сосема нов пристап, нова филозофија во програмирањето. Како основна задача кај овој пристап е претставувањето, на проблемот кој се решава, како модел на реалноста, односно, како систем на интеракции помеѓу објектите кои го сочинуваат моделот. Објектно ориентираното програмирање посебно е погодно за развој на поголеми, комплексни системи. Како логична последица тука се јави потребата да се развие објектно ориентиран модел на информациона систем за проектирање, градба и експлоатација на тунели.

Во трудот е прикажан развојот на тој систем. Најпрво се дадени прегледот на постоечките информациона системи и основните елементи на објектно ориентираното програмирање. Понатаму е прикажана посебно прилагодената методологија за развој на ваков вид информациона системи. Тука е дефиниран проблемот и е извршена глобална анализа на системот. Потоа се утврдени подсистемите и за секој од нив е даден објектно ориентираниот модел. На крајот е опишана имплементацијата на прототип на системот ( во C++), каде што се дадени основните смерници за кодирањето. Извршено е тестирање на прототипот на реален пример.

## 2. ПРЕГЛЕД НА ПОСТОЕЧКИТЕ ИНФОРМАЦИОНИ СИСТЕМИ ЗА ПРИМЕНА ВО ТУНЕЛОГРАДБАТА

При истражувањето т.е. собирањето на податоци за информационите системи кои веќе се користат во тунелоградбата се појавија повеќе клучни моменти кои овде мора да се нагласат.

Прво, тунелите, како градежни објекти, се многу специфични, со многу различни намени, така што е скоро невозможно да се генерализира постапката за проектирање, односно градење. Како последица на тоа имаме отежната примена на компјутери т.е. информациони системи во оваа гранка на градежништвото.

Второ, компјутерската техника и информатиката својот подем го доживува во последните десетина години и аналогно на тоа дури во последните неколку години доживува поширока примена. Како резултат на тоа, посредно, се појавува ситуација да кадрите што треба да ги компјутеризираат процесите (не само во тунелоградбата) дури сега излегуваат од училишните клупи.

Поради овие причини, според досегашните сознанија, не постојат многу разработени информациони системи. Досега повеќе автори имаат работено на различни системи од областа на тунелоградбата. Заедничка одлика на повеќето системи е што се концентрирани (ограничени) на одреден проблем.

Општо, примената на компјутерите во оваа област од градежништвото, се сведува на примена на постоечки програми за анализа на напрегањата во тунелската облога или ридскиот масив (најчесто со примена на методот на конечни елементи). Анализата на предфабрикувани елементи за тунелска облога со помош на методот на конечни елементи и прилагоден компјутерски програм е дадена во [7]. Утврдувањето на реолошките карактеристики на карпите во ридскиот масив околу тунелскиот отвор е дадена во [19]. Применет е методот на конечни елементи, а програмот е прилагоден и за

споредување на резултатите по класичните реолошки теории.

Посебен систем за проектирање на тунели со замрзнување на земјиштето е опишан во [2]. Системот е развиен специјално за тој метод на градба (во слабо - меко земјиште) на тунели и не е применлив за други методи. За градење тунели, исто така, во меко земјиште е развиен систем за *поддршка на одлуките* на инженерите (engineering decision support) [13]. Овој систем посебно е наменет за работа со "штит" машини за пробивање на тунели. Во системот е вградено и одредено инженерско искуство.

Друго поле каде што се имплементирани информациона системи е утврдувањето и следењето на однесувањето на ридскиот масив. Систем за следење на однесувањето на плитките тунели во пескливо земјиште, развиен за утврдување на нов метод за димензионирање на тенки флексибилни тунелски облоги од прскан бетон, е опишан во [1]. За таа цел е изработен лабораториски модел кој е управуван и тестиран со помош на компјутер. Посебно интересен систем е прикажан во [14]. Системот се користи за утврдување на можните критични товари при пробивањето на подземни отвори, а користи неуронски мрежи. Утврдувањето се врши со помош на претходно собрани информации (знаење). Системот може и да собира знаење, од случаите што ги обработува, преку алгоритмот за "учење". Резултатите од "учењето" потоа се користат за испитување на оперативните карактеристики на неуронската мрежа. Системот се разликува од конвенционалните експертни системи, базирани на правила, според начинот на претставување на знаењето и процесот на изнаоѓање решение. За да се обезбедат влезни податоци за слични системи за потребите на управата на Токиското метро е формиран "Geotechnical Data Information System of Tokyo Metropolitan Government" [12]. Самиот систем претставува релациона база за геолошки, геомеханички и геодетски податоци. Системот е поставен уште во 1986 година, а од 1989 година има можности и за графичка презентација.

На крајот мора да го споменеме и системот "ТОНЕЛ" [3] и [4], развиен во институтот "Гидроспетпроект" - СССР. Овој САД систем е наменет за помош при проектирањето на хидро и сообраќајни тунели со профил од 4м<sup>2</sup> до 300м<sup>2</sup>. Системот се базира на стандардите за проектирање на ваков вид објекти, но го користи и искуството од

проектирањето во поранешниот СССР и пошироко. Овде мора да напоменеме дека не станува збор за користење на искуството преку "систем базиран на знаење" туку за вградување на искуството во вид на стандарди за проектирање во процедурите.

Самиот систем се состои од пет подсистеми:

1. проектирање на тунелите, нивните облоги и анализа на профилот;
2. проектирање на рамките за потпирање (ременатите);
3. избор на привремени подградби за тунелите;
4. избор на технологија и
5. утврдување на трошоците.

Потребниот хардвер е IBM PC-AT компатибилен компјутер со плотер. Со помош на овој систем проектирани се повеќе тунели и според искуствата на проектантите постигната е заштеда во време од 60%.

Овде може да заклучиме дека досегашната примена на компјутерската техника и информатика во тунелоградбата се сведува на ограничени, нефлексибилни, решенија. Ова се должи како на специфичноста на тунелите, така и на примената на релативно застарени софтверски решенија (со одредени исклучоци).

Во трудот е направен обид, со користење на современата софтверска техника - објектно ориентираното програмирање, да се генерализира процесот на проектирање и градба на тунелите (во смисла на компјутеризација) преку создавањето на еден флексибилен модел.

Ваквиот објектно - ориентиран модел на информациона систем треба да обезбеди можност за примена на досегашните сознанија за тунелоградбата, како и евентуално вклучување на излезните резултати од другите системи кои веќе се применуваат. Освен тоа треба да биде и лесно проширлив (благодарение на објектно ориентираното програмирање) за нови посовремени технологии во тунелоградбата. Исто така системот ќе треба да овозможи и формирање база на податоци т.е. база на искуство, во која ќе биде вградено искуството од веќе изградените тунели и од тунелите кои ќе

се градат со помош на овој систем. На тој начин ќе се обезбеди можност за користење на тоа искуство при проектирањето и градбата на нови тунели.

## 3. ЕЛЕМЕНТИ НА ОБЈЕКТНО ОРИЕНТИРАНИОТ ПРИСТАП

Во традиционалниот, процедурално - ориентираниот начин на програмирање, програмот опишува серија на чекори кои треба да се извршат т.е. го опишува алгоритмот. Во објектно ориентираниот начин на програмирање програмот го опишува системот на интеракции помеѓу објектите.

### 3.1. ОСНОВНИ КОНЦЕПТИ НА ОБЈЕКТНО ОРИЕНТИРАНОТО ПРОГРАМИРАЊЕ

Објектно - ориентираното програмирање опфаќа неколку клучни концепти. Најосновен од тие е апстракцијата, која овозможува поедноставно пишување на големи програми. Нареден, од концептите, е енкапсулацијата, која го прави полесно менувањето и одржувањето на програмите. На крај е концептот за хиерахија на класите, моќно орудие, кое го прави програмот лесно проширлив. Можно е овие концепти да се применат во било кој програмски јазик, но само објектно - ориентираните јазици ги подржуваат експлицитно [15],[20].

#### 3.1.1. АПСТРАКЦИЈА

"Апстаркцијата" е процес на игнорирање на деталите заради концентрација на битните карактеристики. Додека традиционалните јазици само ја подржуваат апстаркцијата, објектно ориентираните јазици имаат вграден многу по моќен механизам за апстракција[15]. За да се разбере тоа, овде мора да споменеме неколку типа на апстракција:

- **Процедуралната апстракција** е најчеста форма на апстаркција, која подразбира игнорирање на деталите за процесот. Повеќето јазици дозволуваат пишување на програми со највисоко ниво на процедурална апстракција со помош на кориснички

дефинираните функции (процедури или подпрограми). Со пишувањето на такви функции се дефинираат нови термини за изразување т.е. објаснување на работата на програмот. Функциите ги прават големите програми полесни за развој, т.е. за пишување, така што се насочува размислувањето кон логичките информации (ново дефинираните термини), а не кон спецификацијата на наредбите од програмскиот јазик.

- **Апстракцијата на податоците** е друг тип на апстракција, преку кој се игнорираат детали за начинот на претставување на податоците. Ова е поврзано до извесен степен со процедуралната апстракција на тој начин што кога се извршува некоја операција не се води сметка за форматот на податоците. На тој начин се игнорираат детали за дејството на таа операција врз тие типови на податоци. Повеќето програмски јазици се многу ограничени при креирањето на одредени нивои на апстракција на податоците. Карактеристичен пример за апстракција на податоците кај традиционалните јазици се структурите во јазикот "C".

Овде мора да се напомене дека кај традиционалните програмски јазици апстракцијата на податоците и процедуралната апстракција се две различни техники, а кои фактички се поврзани.

- Објектно ориентираните јазици ги комбинираат процедуралната апстракција и апстракцијата на податоци во форма на **класи**: при дефинирањето, на една класа, се опишува се за ентитетот одеднаш. Кога се користи објект од таа класа се игнорираат вградените типови на податоци во класата, како и процедурите кои манипулираат со нив.

Класите претставуваат кориснички дефинирани типови на податоци, структури кои освен податоци може да содржат и функции (членови на таа структура) за манипулирање со тие податоци. Објектите се примероци, инстанци, на соодветните класи. Функциите вградени во класите се нарекуваат "методи". При егзекуцијата на програмот објектите, дефинирани во програмот, ги повикуваат "методите" преку "пораки", кои всушност се интерфејс на класата за поврзување со околината.

Како пример, на овој тип на апстракција, може да се земе едноставната класа "полигон". Полигонот може да се замисли како

серија на точки, а точките може да се замислат како серии на парови од бројки (координати). Но концептуално полигонот е многу повеќе отколку збир на точки или броеви. Полигонот има периметар и површина. Може да се замисли дека сакаме да го движиме или ротираме. Овие карактеристики и операции можат да се добијат односно извршат без повикување на ентитетите на ниско ниво (точките и координатите) кои го сочинуваат полигонот. Значи може да се мисли за полигонот, како објект, без да се мисли на броевите (точките) кои го сочинуваат објектот и без да се мисли на алгоритмите кои манипулираат со него.

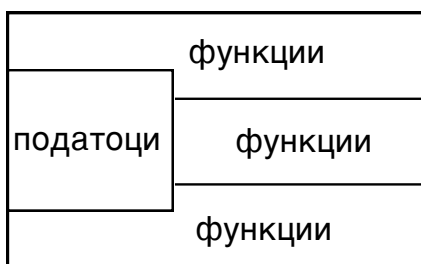
Со развојот на програми засновани на вакви апстрактни ентитети, кои имаат сопствен комплет на операции, се прави програмот независен од деталите на имплементацијата. Ова води кон уште една предност на објектно-ориентираното програмирање : енкапсулацијата.

### 3.1.2. ЕНКАПСУЛАЦИЈА

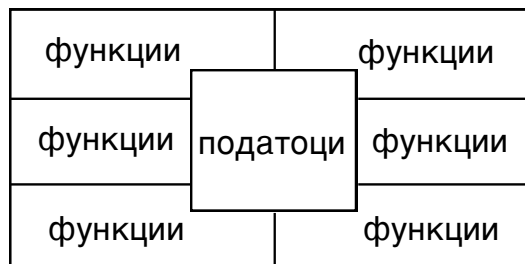
Енкапсулацијата е процес на криење на внатрешните дејства на класата за да се подржи или исфорсира апстракцијата [15],[20]. Ова треба да се сфати како затварање (учаурување) на класата, при што, може да се пристапи кон податоците и кодот само преку "интерфејсот". Ова бара повлекување на реска граница помеѓу "интерфејсот" на класата (кој е со јавен пристап) и нејзината "имплементација" (која има приватен пристап). Интерфејсот опишува што може класата да работи, додека имплементацијата опишува како се работи тоа. На тој начин е подржана апстракцијата со изложување, прикажување, само на релевантните својства на класата: корисникот ги посматра објектите само преку термините - операциите кои тие можат да ги изведат, но не и преку термини за структурата на податоците. Значи овде не станува збор за едноставно комбинирање на функциите и податоците. Можно е поврзување на функциите и податоците во класи со јавен пристап на сите членови, но тоа не е пример за енкапсулација (сл. 3.1.) Вистински енкапсулираната класа ги "опкружува" т.е. ги крие податоците преку своите функции, така да е можен пристап кон податоците само преку повикување на функциите (сл.3.2.).



Енкапсулацијата не се среќава само кај објектно ориентираното програмирање. Принципот на "криење на податоците" во традиционалното, структурно, програмирање истата идеја се применува преку модулите. Општа пракса е да се делат големите програми во модули, кои си имаат јасно дефиниран интерфејс за функциите, така што можат да ги користат други модули. Целта на криењето на податоците е да се направи секој модул колку што е можно по независен.



слика 2.1.



слика 2.2.

На крајот треба уште да споменеме дека енкапсулацијата има повеќе предности. Една од нив е апстракцијата, која е опишана понапред: може да се користи класата (или модулот) без да мора да се мисли за начинот на кој работи. Друга предност е "локацијата", што значи дека промените во еден дел од програмот не бараат промени во другите делови. Слабо лоцираните програми се мошне осетливи на промени во кодот. За разлика од нив програмите со добра локација се многу поедноставни за одржување, а измените делуваат само на мал дел од програмот.

### 3.1.3. ХИЕРАРХИЈА НА КЛАСИ

Една од карактеристиките на објектно ориентираното програмирање, која не може да се најде кај процедуралното програмирање, е можноста за дефинирање на хиерархија на типови [15],[18],[20]. Тука станува збор за дефинирање на една класа како подтип, односно посебна категорија, на друга класа т.е. едната класа да се изведе од другата. Исто така може да се изразат сличностите помеѓу класите, односно да се дефинираат како подкатегории на една поширока категорија, со нивно изведување од една основна класа.

Идентификувањето на општа основна класа за повеќето класи е форма на апстракција: Основната класа е претставеник на тие класи на највисоко ниво. Таа ги одредува најопштите карактеристики на изведените класи, така да може да се сконцентрираме на тие доделени особини и да ги игнорираме нивните индивидуални карактеристики. Оваа техника на апстракција овозможува да се гледаат ентитетите (термините) преку мал број на категории. Оваа техника се користи и во секојдневните комуникации; на пример, полесно е се мисли "цичачи" отколку "лавови, тигри, мечки ...", или пак "мечки" наместо "гризли, црна мечка, поларна мечка ...".

Така основната класа е **генерализација** на група од класи. Аналогно, изведената класа е **специјализација** на некоја друга класа; Изведената класа е подтип на претходно препознаен тип и го опишува со термини за неговите додатни карактеристики. На пример, лавовите се цичачи, но тие исто така имаат и некои особини кои не се среќаваат кај сите цичачи.

Постојат две практични придобивки од дефинирањето на класната хиерахија: изведената класа може да го добие кодот од основната класа, или пак да го добие интерфејсот од основната класа.

- **Наследување на кодот:** Ако имплементираме класа и сакаме да ја искористиме функционалноста на некоја постоечка класа, едноставно може да изведеме класа од постоечката. Ова е ситуација кога наследувањето дозволува повторна употреба на кодот. На тој начин ако имплементираме повеќе класи со заеднички карактеристики, класната хиерахија може да го намали редувантното кодирање. Значи општите особини ќе се имплементираат само еднаш во основната класа и нема да се повторуваат во секоја изведена класа.

Класната хиерахија развиена со повторна употреба на кодот, најголем дел од кодот го има во основната класа (на врвот од хиерахијат). На тој начин кодот може да се употреби од многу класи. Така изведените класи претставуваат специјализирана, односно проширена верзија на основната класа.

- **Наследување на интерфејсот:** Друга стратегија за наследување е ако изведената класа ги наследи само имињата на функциите-членки на основната класа. Кодот во овој случај не се наследува:

изведената класа има свој сопствен код за тие функции. Значи изведената класа го има истиот интерфејс како и основната класа, но различни особини на истите функции. Оваа стратегија овозможува различни класи да можат да користат ист интерфејс, со тоа што се зголемува нивната сличност на повисоко ниво. Најголема придобивка од наследувањето на класите е полиморфизмот (можноста да се повикуваат функциите - членки од некој објект, без да се специфицира точниот тип на објектот е позната како полиморфизам - можност да се добијат многу форми).

Во пракса најчесто се применува комбинација на обете стратегии на наследување но можно е и стриктна примена само на наследување на кодот или пак само на наследување на интерфејсот. Во вториот случај во основната класа се декларираани, односно утврдени, само функциите, но не е имплементиран и кодот. Овде станува збор за апстрактна основна класа.

Класната хиерархија развиена за наследување на интерфејсот најголем дел од кодот содржи во изведените класи (на дното од хиерархијата). Во овој случај изведените класи претставуваат работна верзија на апстрактниот модел дефиниран со основната класа.

....

Класите можат да подржуват апстракција, енкапсулација и хиерархии. Класите се механизам за дефинирање на апстрактни типови на податоци преку кои се извршуваат операциите. Класите може да бидат енкапсулирани, разделувајќи го кодот и нагласувајќи ја неговата локација. На крај, класите може да бидат организирани во хиерархии, нагласувајќи ја меѓусебната поврзаност и истовремено минимизирајќи ја редундантноста на кодот.

*За да се добие најмногу од објектно ориентираното програмирање не е доволно само да се напише програм т.е. да се имплементира одреден систем, во некој објектно ориентиран јазик. Тежиштето треба да се постави на развојот на моделот на системот.*

## 3.2. РАЗВОЈ НА ОБЈЕКТНО-ОРИЕНТИРАНИ СИСТЕМИ

Објектно-ориентираното програмирање претставува нов начин на мислење, кој фундаментално се разликува од начинот на мислење при класичното програмирање [16],[21].

Во top-down структурното програмирање првиот чекор е да се одреди целта на програмот. Тоа значи дека прво се опишуваат главните чекори кои мора да ги изведе програмот (со користење псеудо кодови или дијаграми). дури потоа главните чекори се делат на помали со што се допрецизира функцијата на програмот. Оваа техника е позната како процедурална (функционална декомпозиција и го третира програмот како опис на процес, кој понатаму се дели на потпроцеси.

Објектно ориентирано развој драстично се разликува од оваа техника. Во објектно ориентирано развој не се анализира проблемот преку задачите што треба да ги извршува. Како приоритетна задача овде се јавува утврдувањето на објектите и нивните врски.

Друга битна разлика помеѓу објектно ориентирано начин на програмирање и структурното програмирање е што објектно ориентирано развој вклучува работа и на високо и на ниско ниво во сите степени на развој на системот.

Самата постапка за развој на објектно-ориентираните програми е многу флексибилна, се состои од следните чекори [20],[25]:

- идентификација на класите (објектите);
- доделување на атрибути и методи на класите;
- утврдување на врските помеѓу класите;
- уредување на класите во хиерархии.

Овде мора да се напомене дека развојот на објектно ориентираните програми е итеративен процес: секој чекор во процесот може да ги измени претпоставките од претходниот чекор, што понатаму бара враќање назад и повтарање на чекорот со новите информации.

Горе наведените чекори може да се реализираат на различен начин, во зависност од проблемот кој се решава. Понатаму, во наредниот дел, ќе биде објаснета применетата постапка за развој на конкретниот објектно ориентиран систем.

## 4. МОДЕЛ НА ИНФОРМАЦИОНИОТ СИСТЕМ

### 4.1. МЕТОДОЛОГИЈА ЗА РАЗВОЈ НА ИНФОРМАЦИОНИОТ СИСТЕМ ЗА ПРОЕКТИРАЊЕ, ГРАДБА И ЕКСПЛОАТАЦИЈА НА ТУНЕЛИ

За утврдување на моделот на информациониот систем користена е следнава постапка [24]:

1. Најпрво е дефиниран проблемот, при што прецизно се утврдени задачите кои треба да бидат решени при развојот на системот. Исто така утврдени се сите релевантни податоци за еден ваков систем. Паралелно со дефинирањето на проблемот за подобро следење на текстот дадена е карактеристичната терминологија за тунелоградбата која е илустрирана со скици на одредени елементи на тунелите.
2. Понатаму е извршена глобална анализа на системот и при тоа се утврдени основните подсистеми. Овде се одредени и пристапите врз основа на кои ќе се врши понатамошно решавање на системот, а тоа се: објектно ориентирано програмирање со користење на активни бази на податоци.

Глобалниот модел е претставен и шематски.

3. Во третиот чекор разгледувани се поединечно секој подсистем и тоа во два чекора:
  - анализа на подсистемите и
  - утврдување на објектно ориентираниот модел на подсистемот.
- 3.1. За да се формализира моделот на реалноста се започнува со замислување т.е. поставување на дијаграм на ток на податоци - DFD (Data Flow Diagram). На тој начин со анализата на подсистемите е извршена детална функционална декомпозиција [6].

Врз основа на оваа анализа се прецизирани задачите на секој подсистем, и се дадени преку DFD Тука е утврден токот на податоците, влезот и излезот т.е. трансформацијата на податоците како и врските помеѓу различните трансформации. Символите кои се користени се:

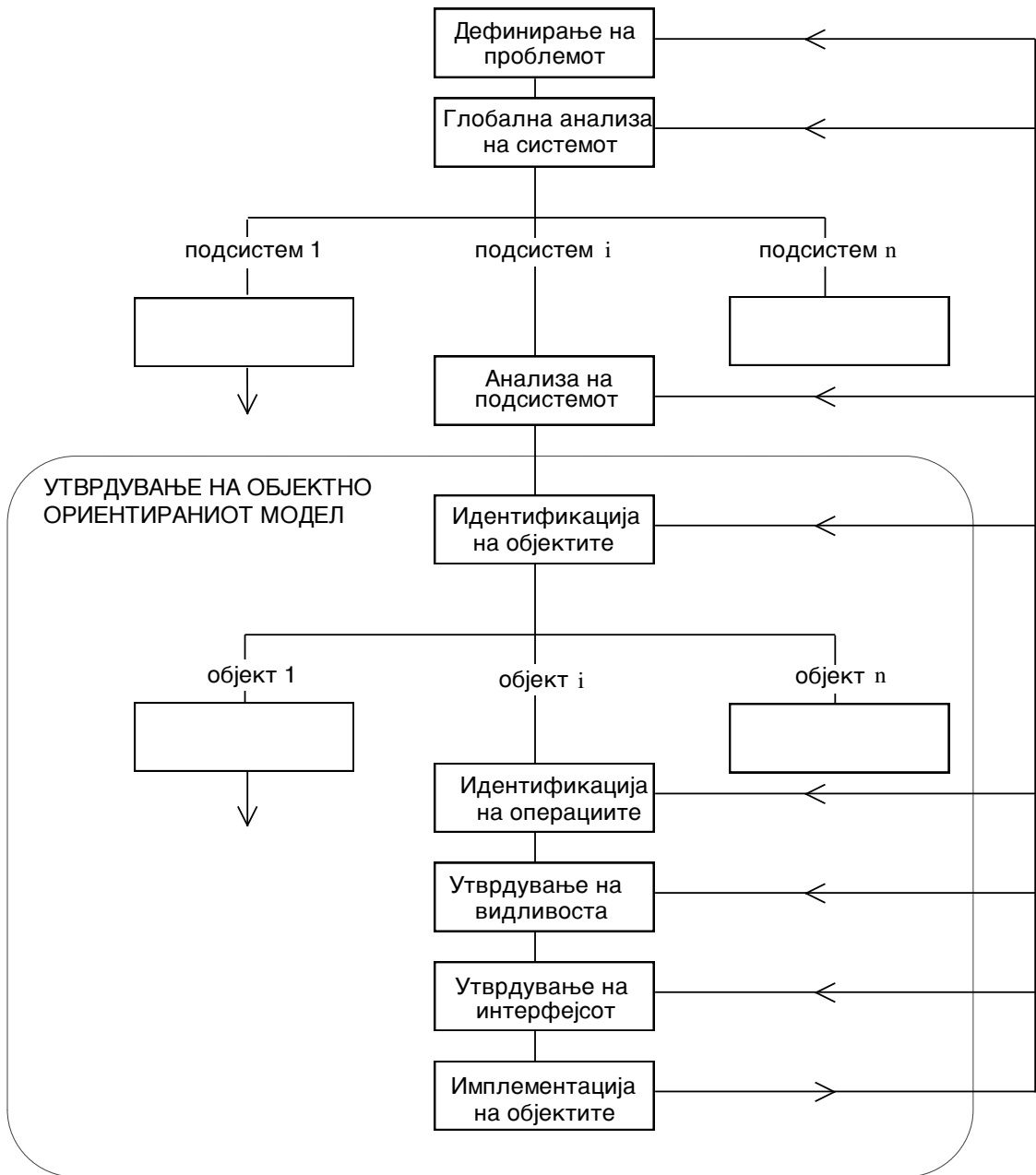
- правоаголник - претставува надворешен ентитет (надвор од границите на системот)
- стрелка - го покажува токот на податоците
- круг - ја претставува трансформацијата на податоците
- две паралелни линии - ги претставуваат податоците кои се чуваат (кои може отпосле да се читаат)

3.2. На крајот врз основа на DFD утврден е објектно ориентираниот модел според следнава постапка:

- Идентификација на објектите и нивните атрибути;
- Идентификација на операциите кои ги претрпува и се бараат од секој објект;
- Утврдување на видливоста на секој објект во релација со другите објекти
- Утврдување на интерфејсот за секој објект;
- Имплементација на објектите.

4. Како последен чекор е предвидено тестирање на прототипот каде што ќе се утврди квалитетот и валидноста на системот.

Целокупната постапка е прикажана шематски (сл. 4-1), каде што секој чекор е означен како повратен. Тоа значи дека е можно враќање на некој од претходните чекори при развојот на системот со што по извршени одреден број итерации ќе се добие квалитетен систем. Ова произлегува од таму што моделирањето на било кој систем претставува креативна работа и е можно да се реши на повеќе начини. Тоа значи дека првобитните замисли може во понатамошната работа да не се покажат како добри решенија, што повлекува враќање на некој претходен чекор, па дури и враќање на дефинирањето на поставената задача.

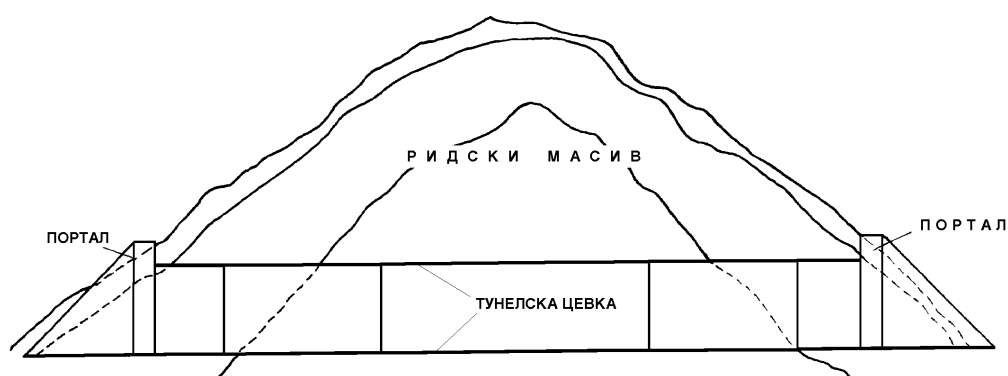


Слика 4-1 Моделирање на информациониот систем



## 4.2. ДЕФИНИРАЊЕ НА ПРОБЛЕМОТ

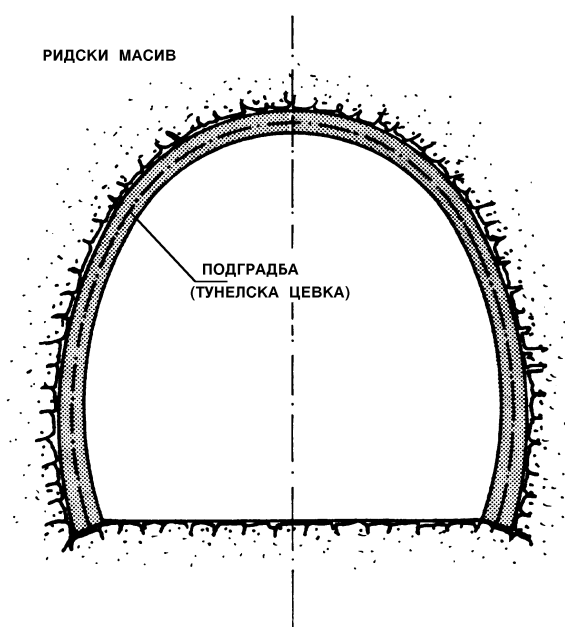
Тунелите се инжењерски подземни градби со чија помош се совладуваат теренски препреки при изградбата на сообраќајници, хидро-системи, енергетски постројки, а може да служат и како скривници или магацини [16].



слика 4-2 Надолжен пресек на тунел

Градбата на било каков градежен објект се изведува по точно утврдена процедура, односно технологија. Се почнува со изработка на инвестиционата документација, а потоа се преминува на проектирањето т.е. на изработка на проектната документација. Дури потоа може да се помине на изведувањето (физичката реализација - изградба) на самиот објект. Во зависност од видот на објектот можно е со помала или поголема прецизност да се одвои процесот на проектирање од процесот на изградба. Тунелите, како инженерски објекти, спаѓаат во групата каде што не е можно да се повлече реска граница помеѓу проектирањето и изградбата. Тоа се должи на фактот што во процесот на проектирање се влегува со многу непознати за видот и квалитетот на карпите (ридскиот масив, геологијата, сл. 4-2) во кои треба да се пробива тунелот. Можно е многу попрецизно утврдување на карактеристиките на карпите, што секако си има и своја цена која не е за потценување. Меѓутоа и во тој случај,

карактеристиките на карпите треба да се примаат со резерва поради накнадното дејство од експлозивот (при пробивањето на тунелот) или поради непредвидените реакции на карпите при изложување на дејство на воздух (или вода). Поради тоа при проектирањето се оди со прогнозен геолошки профил врз основа на кој се врши димензионирање на тунелската подградба т.е. цевка (слика 4-3) и тоа на тој начин што се утврдуваат повеќе различни типови на конструкција, а според тунелските (подземните) притисоци што можат да се очекуваат.

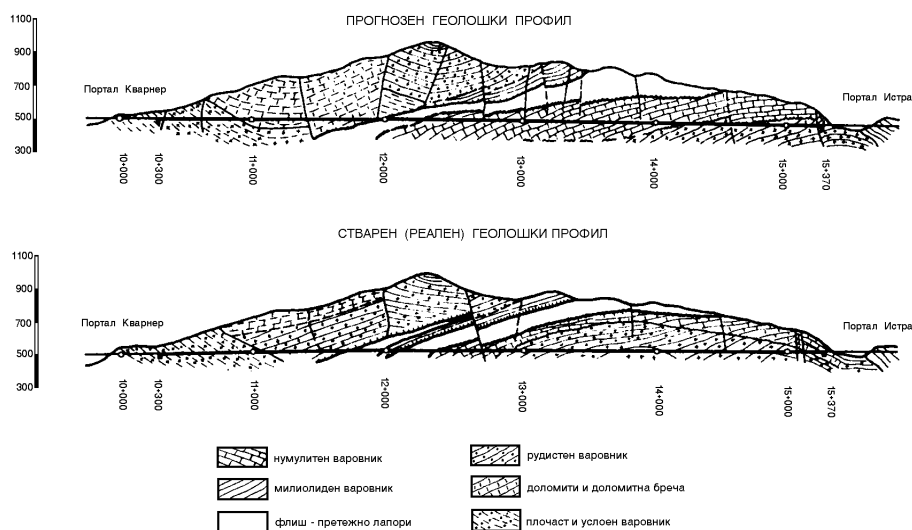


слика 4-3 - Најречен пресек на тунел

Конечното димензионирање се врши на лице место со утврдување на реалната геолошка состојба. При тоа можни се две ситуации: реалната геологија да се сложува со прогнозниот геолошки профил или да не се сложува (слика 4-4). Ако геологијата се сложува тогаш се вградува предвидениот тип на тунелска цевка, а ако не се сложува тогаш се бара тип кој одговара на реалната геологија. Тука, повторно, можни се две ситуации: бараниот тип да постои во проектот или бараниот тип да мора да се проектира (пресметува) на лице место - што претставува додатно оптеретување и за проектната екипа и за изведувачите. Порано постапката на конечното димензионирање ја вршела посебна стручна комисија, но со развојот на науката и техниката усовршени се разни мерни инструменти кои можат

поконкретно да ги дадат карактеристиките на карпите. Поради тоа се јавува потреба од систем кој ќе ја следи состојбата на ридскиот масив при пробивањето на тунелите и ќе препорачува, односно, ќе нуди решенија за одредени ситуации (на пример ќе утврдува кој тип на подградба треба да се вгради или пак ќе врши проектирање на нов тип).

На ова се надоврзува и проблемот со технологијата на изградба т.е. методот по кој ќе се пробива тунелот. Според класичните методи тунелската цевка е со точно одредена димензија и не е оставен простор за интервенција. Со современите методи конечното димензионирање се врши со постапно вградување на дел од подградбата и активно следење на напрегањата и деформациите, во ридскиот масив и самата подградба, преку посебни инструменти. При тоа одредени се критичните вредности на напрегањата (или деформациите) при кој треба да се вгради наредниот дел од подградбата (најчесто додатни анкери или слој прскан бетон). Значи, сега станува збор за еден друг, паралелен, систем кој треба, освен да ја следи состојбата во ридскиот масив, и да го следи однесувањето на конструкцијата и благовремено да праќа сигнали за интервенции. Една од методите која може да го применува тој систем е "Новата Австриска Метода" за градба на тунели [22],[23].



слика 4-4 Прогнозен и стварен геолошки профил (тунел Учка - Хрватиска)

Освен тоа, во постапката на градба на тунелите, почнувајќи од проектирањето, преку пробивањето т.е. изградбата на тунелот, па се до пуштањето во експлоатација, многу повеќе се потпираме на емпириски методи отколку на пресметувања по строги физички закони (поради што голем број на автори зборуваат за "вештина на градењето тунели" наместо за "наука за градење тунели"). Ова доаѓа до израз во прв ред при прогнозирањето на тунелските притисоци како и при одредувањето на реакциите на масивот при пробивањето на тунелите. Со цел да може да се искористи искуството од проектирањето и градењето на тунелите пожелно е да се оформи база на податоци во која ќе биде складирано искуството и знаењето од конкретните изведени проекти.

Значи како главни задачи на идниот информационер систем се појавуваат:

- проктирање на тунелите; прогнезен геолошки профил и типови на подградба
- следење на состојбата во ридскиот масив и споредување на реалната со проектираната геологија како и препорачување на тип на подградба;
- следење на напрегањата, притисоците и однесувањето на ридскиот масив, како и праќање сигнали за евентуална интервенција;
- формирање база на податоци за подоцнежнo користење (како искуство) при проектирање и изградба на нови тунели.

При дизајнирањето на информационер систем кој би ги задоволувал горе наведените аспекти најбитно е да се утврди постапката на градба на тунелите, како и да се утврдат релевантните податоци за валидно и компетентно претставување на комплетниот процес.

Како што веќе погоре е споменато постапката за градба на тунели може да ја претставиме во три фази:

- фаза проектирање;
- фаза изградба и
- фаза експлоатација.

Првата фаза, во глобала, може да ја претставиме преку неколку последователни чекори:

1. претходни истражувања и рекогносцирање на теренот (геодетски снимања, геолошки и хидрогеолошки истражувања и сл.);
2. дефинирање на тунелот т.е. одредување на параметрите на тунелот: должина, профил, нивелета итн.);
3. прогнозен геолошки профил со класификација на ридските материјали (геологија во која се пробива тунелот) и утврдување на меродавните физички големини;
4. утврдување на подземните (тунелските) притисоци (според прогнозниот геолошки профил);
5. димензионирање на подградбата односно тунелската облога (цевка).

Третиот и четвртиот чекор се однесуваат на прогнози поради тоа што стварната состојба во ридскиот масив, фактичката геологија, се утврдуваат на лице место при самото пробивање на тунелот. Колку пореални ќе бидат овие прогнози зависи од средствата и напорите вложени во геолошките истражувања и во колкава мера ќе бидат искористени искуствата од претходните градби. Во врска со утврдувањето на прогнозните тунелски притисоци, а во зависност од условите, се применуваат повеќе општо прифатени теории кои за ова ниво на разработка на системот не би требало да го оптеретуваат текстот. Овде само треба да се напомене дека овие теории и за исти услови т.е. исти влезни податоци даваат во голема мера различни резултати. Геолошката прогноза и прогнозата на притисоците би требало да се разработат во посебни подсистеми базирани на знаење при понатамошната разработка на овој систем.

Фазата "изградба" всушност е најбитна за целиот систем. Во текот на оваа фаза се утврдува реалната геологија и стварните тунелски притисоци и се споредуваат со проектот. За градба на тунели познати се доста различни методи, но како најприменувана и најекономична во последниве неколку десетина години се покажала "Новата Австриска Метода". Накратко суштината на оваа метода се состои во тоа да се искористи во максимална можна мера носивоста на ридскиот масив над пробиениот тунелски отвор. Со постапно додавање на подградба, која се состои од анкери и армиран прскан бетон, се формира спрегнат носив слој од подградбата и дел од ридскиот масив. Самата метода, упростено, изгледа вака:

- избивање на профилот (може да биде во повеќе нивои);
- утврдување на лице место дали проектираната (прогнозната) геологија се совпаѓа со реалната. Ако има разлики се преработува проектот и се утврдува потребната подградба т.е. се врши конечното димензионирање на подградбата;
- поставување на мерни анкери, екстензиометри или мерни перници за следење на состојбата на напрегање во ридскиот масив над отворот;
- контрола (мерење) на деформациите или притисоците и соодветни интервенции;
- постапно поставување на дел од проектираната подградба и контролирање на деформациите и напрегањата и во ридскиот масив и во подградбата.

Постапката на мерење на деформациите или напрегањата на овој степен на развој на тунелоградбата се врши мануелно, но возможно е и поврзување на мерните инструменти со сензори така што ќе биде овозможено "on line" т.е. автоматско мерење со комплетна обработка на податоците преку некој, на пример, персонален компјутер. Вака добиените податоци се бележат и се споредуваат со проектираните. Кога добиените вредности ќе дојдат до критичната точка системот треба да алармира, а поставената подградба се појачува и повторно се врши контрола до моментот кога ќе се утврди дека деформациите или напрегањата не се зголемуваат т.е. се смируваат. Како критична точка се означува моментот кога се надминуваат дозволените напрегања во подградбата односно во масивот.

Третата фаза "експлоатација" всушност не е активна "работна" фаза и нејзината улога се состои во тоа што редовно треба да се следи однесувањето на тунелот преку мерни инструменти, на ист начин како и при изградбата. Целта е да се добие потврда дека тунелот е правилно изграден или пак ако се појават неповолни мерења системот треба да даде сигнал за благовремено да се интервенира со соодветна санација. Освен оваа санација, друга придобивка на ова следење на ситуацијата е што искуството може да се пренесе и на други објекти со слични услови. Фазата "експлоатација" може условно да се сфати: не се мисли на буквалното предавање на тунелот на користење туку се мисли на комплетирањето (бетонирањето) на тунелската цевка. Поради тоа што тунелите се линиски објекти најчест случај е да на одредена

должина тунелска цевка биде комплетна додека сеуште се врши ископ на челото. Тука мора да се спомене и временската димензија: најчесто станува збор за мерење на времето на изградба во години.

Ова би го поткрепил со примерот за изградба на тунелот Златибор на пругата Белград-Бар со должина од 6169,30 метри, а чија изградба траеше од 1961-70 година. Ова е, истовремено, и конкретен пример за конечно димензионирање на тунелската цевка и за користење на искуството од градбата на еден тунел при понатамошната градба на истиот тунел со видливи позитивни резултати, но за жал просторот овде не дозволува задржување и разработка.

Релевантните податоци со кои може да се опише еден тунел можат да се поделат во неколку групи:

1. податоци со кои глобално се опишува проектот:
  - име на тунелот, назив;
  - намена;
  - краток текстуален опис;
  - должина;
  - профил и сл.
2. податоци за геологијата (може да има повеќе различни класи):
  - опис на карпата;
  - слоевитост;
  - испуканост;
  - влажност;
  - кохерентност;
  - јакост и
  - други физички карактеристики.
3. податоци за типовите подградби (може да ги има повеќе типови, во зависност од бројот на различните геологии):
  - опис на подградбата т.е. елементи на подградбата;
  - во која класа т.е. во која геологија може да се примени;
  - гранични вредности на притисоците за кои може да се применува;
  - максимална деформација која може да ја поднесе;
4. податоци со кои поконкретно се опишува тунелот - податоци за напречните профили (коти, геологија, тип на подградба итн.):

- стационача;
- кота на нивелета;
- кота на терен;
- класа на карпа - геологија;
- тип на подградба.

Освен овие ќе треба да се располага и со :

5. податоци за поставените мерни профили:

- стационача на која се поставува мерниот профил;
- проектирана т.е. прогнозирана класа на карпа;
- реална класа на карпа;
- датум и време на избивање на профилот;

6. податоци за извршените мерења:

- на кој профил е извршено мерењето;
- проектирана подградба;
- вградена подградба;
- временски распоред на мерењата;
- гранични вредности на мерењата (од типот на подградбата);
- измерени вредност.

На крај потребно е уште да се напомене и дека сите овие податоци во секое време треба да биде можно да се прегледат т.е. да се користат.



### 4.3. АНАЛИЗА НА СИСТЕМОТ СО ГЛОБАЛЕН МОДЕЛ

Појдовната цел - градба на тунел - се извршува, како што веќе видовме, во неколку фази. Најпрво потребно е да се изработи проект со сите нумерички и графички прилози, а потоа се преминува на изградбата на самиот тунел. Поради специфичноста на тунелите, како градежни објекти, како трета фаза пожелно е да се земе фазата "експлоатација" на тунелот каде што доаѓа предвид следењето на конструкцијата и соодветно на тоа одржување на конструкцијата.

Значи ќе се јават три информациона подсистеми кои ќе можат да работат независно:

- ИПС "Проектирање"
- ИПС "Градба"
- ИПС "Експлоатација"

Освен тоа овде се појавува и систем за управување со базата на податоци DBMS, како и систем за управување со правилата RuMES, кои заедно го чинат системот за управување со активна база на податоци ADBMS.

Подсистемот "проектирање" како влез ги зема претходните испитувања т.е. геодетските податоци и податоците од геолошките испитувања. Исто така како влезни податоци во овој подсистем може да се јават и податоците од веќе изградени тунели (кои ќе се користат како претходни искуства и кои се наоѓаат складирани во базата на податоци). Како излез треба да се добие комплетен опис на тунелот со сите, веќе споменати, релевантни податоци. Овде е важно уште еднаш да се напомене дека како конечни податоци се јавуваат типовите на подградба за тунелот, додека распоредот на геологијата и распоредот на самите типовите на подградбата се прогнозирани и реалните податоци се утврдуваат на лице место при пробивањето на тунелот т.е. со конечното димензионирање. Тоа е всушност и улогата на вториот подсистем "градба".

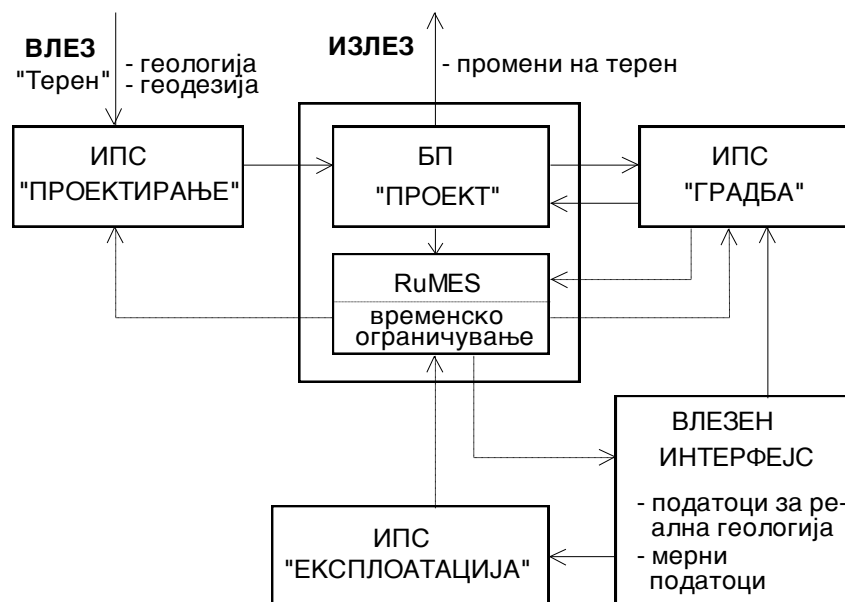
Подсистемот "градба" ги користи податоците од проектот, како и податоците за геологијата добиени на лице место при пробивањето на тунелот. Кога ќе се добијата реалните податоци системот го повикува

RuMES и ги активира потребните правила. Во случај ако реалната геологија не се совпаѓа со проектираната (прогнозираната) системот треба да побара ажурирање на проектот. Втората улога на овој подсистем е следењето на деформациите (односно напрегањата) во тунелот, а за кои треба да се активира пак RuMES со друга група на правила. Отчитаните вредности се споредуваат со проектираните и соодветно на тоа се бара вградување на одреден тип на подградба. Отчитувањата се вршат по однапред утврден временски распоред.

На крај откако ќе се заврши тунелот (може и дел од тунелот) се активира третата фаза: "експлоатација". Овде во тек на време се контролираат деформациите (односно притисоците) на тунелската цевка и соодветно на тоа се бара или не се бара интервенција.

Подсистемите "градба" и "експлоатација" располагаат и со влезен интерфејс преку кој се отчитуваат податоците за реалната геологија и измеренерените вредности преку кои се следи однесувањето на тунелот. Овој интерфејс се активира на барање на корисникот или пак на одреден временски интервал. Временскиот интервал по правило би требало да го определува системот според однапред утврдени услови (дефинирани во подсистемот "проектирање").

Од горе наведеното се гледа дека системот работи како целина од три независни подсистеми кои се поврзани преку (т.е. заеднички им е) ADBMS. шематски тоа би изгледало како на слика 4-5.

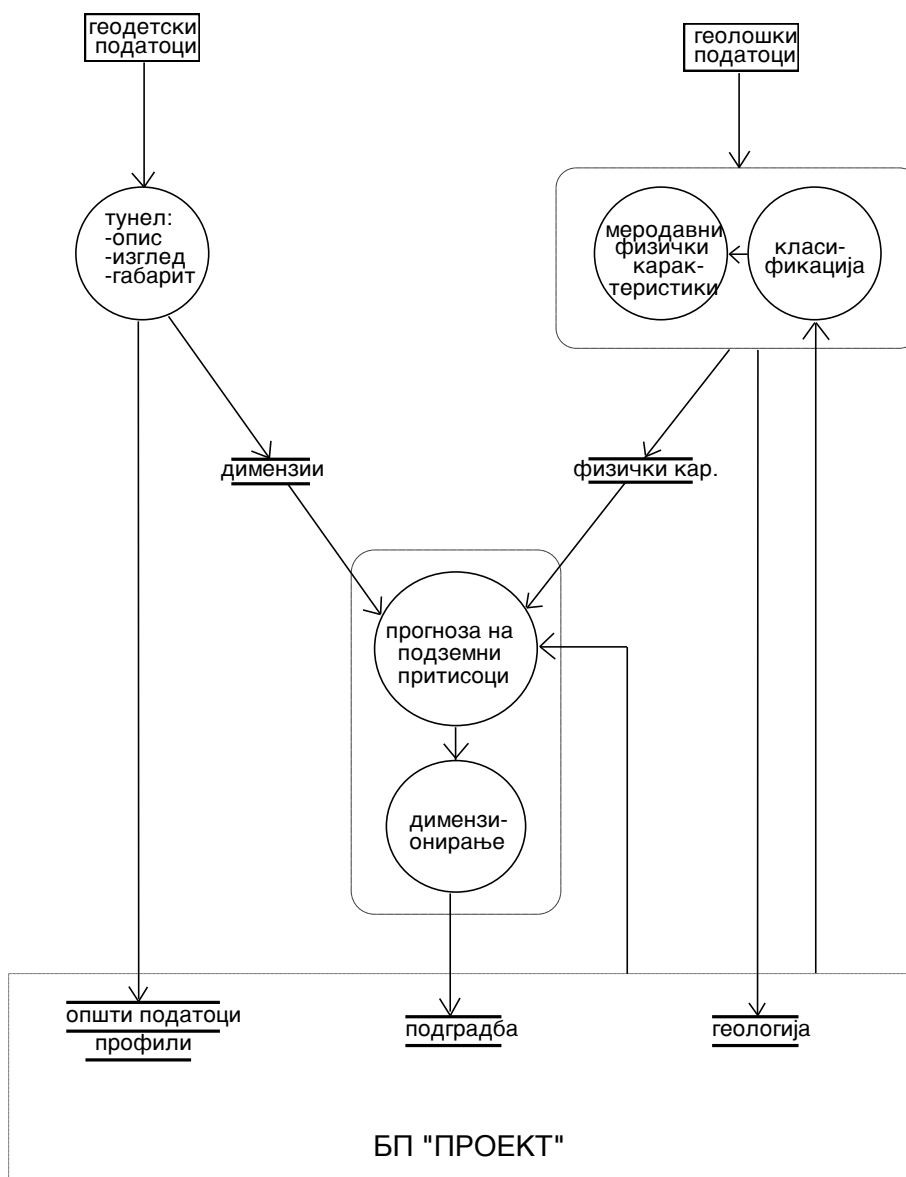


Слика 4-5 Глобален модел на системот

## 4.4. ИНФОРМАЦИОНЕН ПОДСИСТЕМ "ПРОЕКТИРАЊЕ"

### 4.4.1. АНАЛИЗА НА ПОДСИСТЕМОТ

За да се формализира моделот се започнува со замислувањето на токот на информациите, кој се претставува преку еден вид DFD (data flow diagram ) на слика 4-6.



Слика 4-6 Информационен подсистем "Проектирање"

Со понатамошната, детална, функционална декомпозиција утврдено е дека ИПС "Проектирање" се состои од четири различни подсистеми (модули):

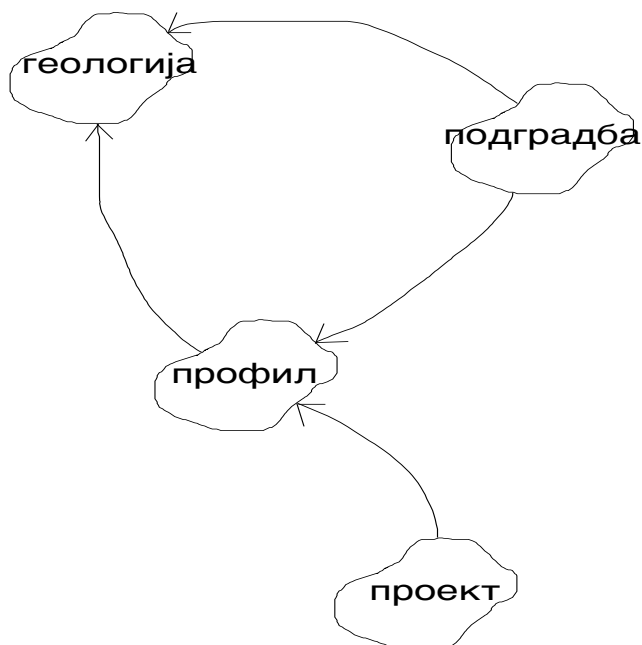
- модул за внесување на основните податоци за тунелот;
- модул за внесување на податоците за геологија, класификација и утврдување на меродавните физички карактеристики на карпите;
- модул за утврдување на подземните притисоци;
- модул за димензионирање на подградбата (тунелската цевка).

Сите овие подсистеми можат да работат, до извесна мера, независно и можно е нивно понатамошно разработување. За ова ниво на изградба на системот вградени им се само најосновните функции, но при разработката на системот на повисоко ниво неопходно би била дополнителна работа. Врската помеѓу овие подсистеми е дадена на сликата 4-6

ИПС "Проектирање" како влезни податоци ги прима геометријата на тунелот и геодетските податоци за теренот, како и прогнозните геолошки податоци за составот на карпите. Неговата задача е да изврши класификација на геологијата, да ги издвои меродавните физички карактеристики на карпата, врз основа на геометријата на тунелот и меродавните физички карактеристики на карпата да даде прогноза на меродавните тунелски притисоци. Класификацијата на геологијата и прогнозата на подземните притисоци се врши со постојана консултација со базата на податоци при што се користи искуството од веќе изградените тунели. На крајот со помош на така утврдените притисоци се врши димензионирање на подградбата. Сите валидни податоци се запишуваат во базата на податоци за примена во другите фази. На тој начин како излез од овој подсистем е добиен комплетниот опис на тунелот приготвен за изградба. Овој опис понатаму треба да се ажурира во зависност од реалната ситуација при пробивањето на тунелот.

#### 4.4.2. ОБЈЕКТНО ОРИЕНТИРАН МОДЕЛ НА ИПС "ПРОЕКТИРАЊЕ"

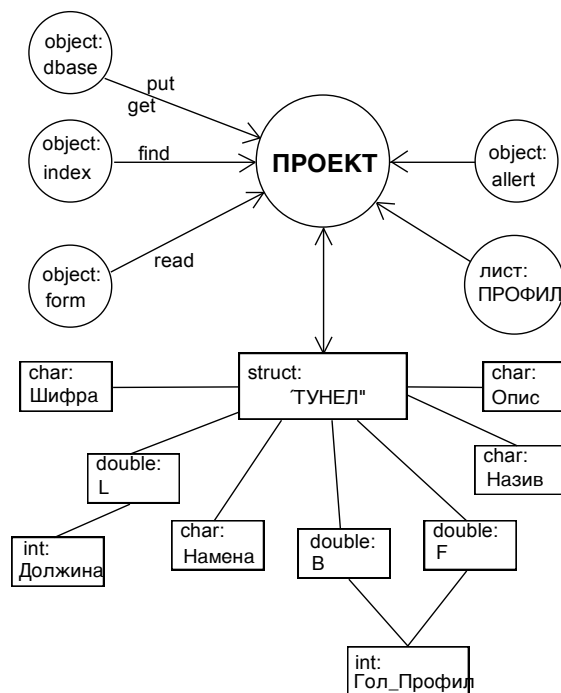
При дизајнирањето на објектно-ориентираниот модел на реалноста мора да се има предвид дека апстракцијата и криењето на податоците ја формираат основата на објектно-ориентираниот развој. Основен критериум за декомпозиција на системот кој ги користи објектно-ориентираните техники е: *секој модул во системот означува објект или класа на објекти од областа на проблемот.*



Слика 4-7 Објекти на подсистемот 'Проектирање'

Првиот чекор, *идентификацијата на објектите и нивните атрибути*, го опфаќа препознавањето на главните актери, агенти и сервери од областа на проблемот како и нивната улога во моделот на реалноста. Во овој подсистем се идентификувани конкретни објекти (сл. 4-7), кои, вообичаено, се изведуваат од опишаните области на проблемот. Како основа за идентификација служи дијаграмот од слика 4-6, а се поаѓа најпрво од основните ентитети. Во овој подсистем како основни, независни, ентитети се појавуваат геологијата (вид и квалитет на карпата) и подградбата. Овие ентитети всушност се првите идентифицирани објекти. Може да ги има повеќе примероци, но само по еден одреден примерок се составен дел на напречниот профил, даден овде како ентитет - односно како објект. Преку напречните профили се дефинира комплетно тунелот. Секој

напречен профил освен од опис на геологијата и подградбата (објекти) се состои и од други карактеристични податоци што го опишуваат тунелот како проект (стационажа, неопходни коти). Прецизноста со која ќе се претстави тунелот зависи од бројот на објектите "напречен профил". Значи, на крајот, како објект кој ги обединува претходните објекти, се појавува објектот "проект", кој се состои од извесен број "напречни профили", како и од податоци кои генерално го опишуваат тунелот.



Слика 4-8 Ајрибуџи на објектиот "Проект"

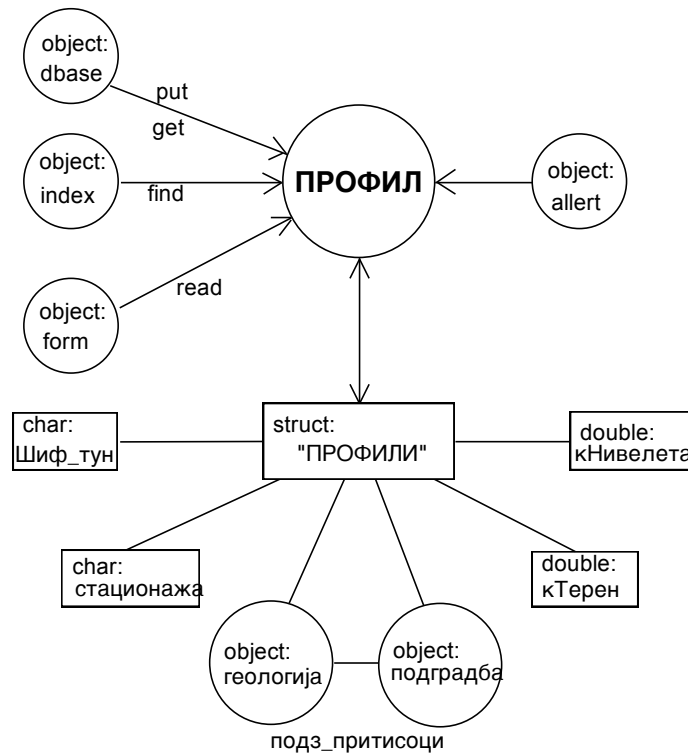
Објектите (на ова ниво на декомпозиција) треба да ги содржат следниве податоци:

- "проект" - опис на тунелот - ги содржи основните податоци за тунелот: намена, опис, назив, должина, профил . . . (сл.4-8);
- "профил" - напречни профили - со помош на профилите најдетално се опишува тунелот - овде се дадени стационажата, конкретниот надслој, геологијата и подградбата (сл.4-9);

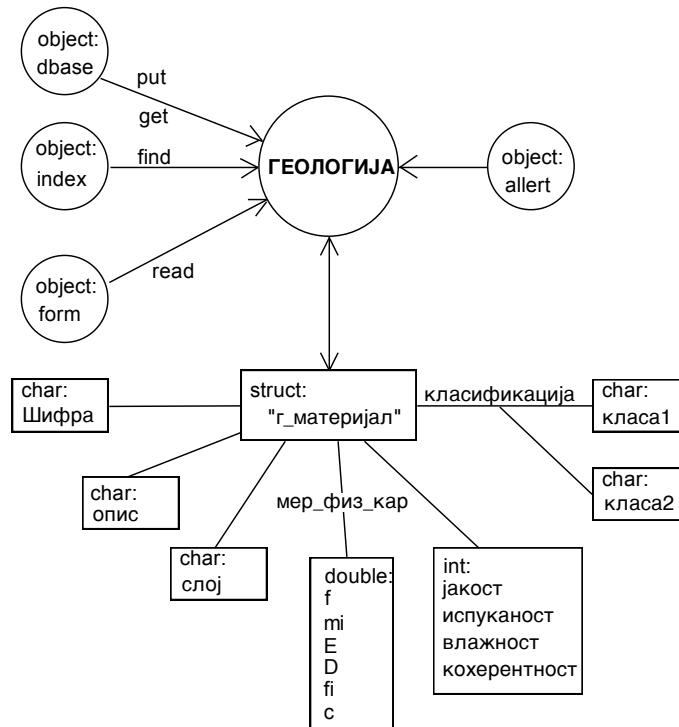
- "геологија" - ги содржи податоците за материјалот во кој ќе се гради тунелот. За еден тунел може да има повеќе типови на материјал (сл.4-10);
- "подградба" - зависи од геологијата и надслојот (сл.4-11), а го содржи описот на подградбата: во која геологија се применува, за кои притисоци е димензионирана, после колку часа се се вградува.

Како дел од атрибутите на гореспоменатите објекти се јавуваат и библиотечните објекти (класи):

- database - овозможува запишување и читање од диск
- btree - овозможува индексирање и пребарување на датотеките.
- dialogue - прозор (форма, маска) кој служи како кориснички интерфејс за работа со податоците.
- alert - дел од корисничкиот интерфејс кој служи за информирање или опоменување на корисникот за однесувањето на системот во врска со одреден објект.



Слика 4-9 Атрибути на објектот "Профил"



Слика 4-10 Атрибути на објектот "Геологија"

Како посебен атрибут е дадена структура која ги содржи податоците, односно останатите атрибути, за секој објект. Ова не е неопходно, но поради практични причини (за да се поедностави запишувањето и пребарувањето на податоците) е усвоено ова решение.

Наредниот чекор, *идентификацијата на операциите кои ги претрпуваат и се бараат од објектите*, е постапка која го карактеризира однесувањето на секој објект или класа објекти. Овде се утврдува статичката семантика на објектите со одредување на податоците и операциите кои се значајни т.е. се однесуваат на објектот или за објектот. Во овој чекор би требало да се утврди и динамичкото однесување на секој објект преку идентификацијата на ограничувањата, во однос на време или простор. На овој начин овозможено е одвојување на објектите едни од други.

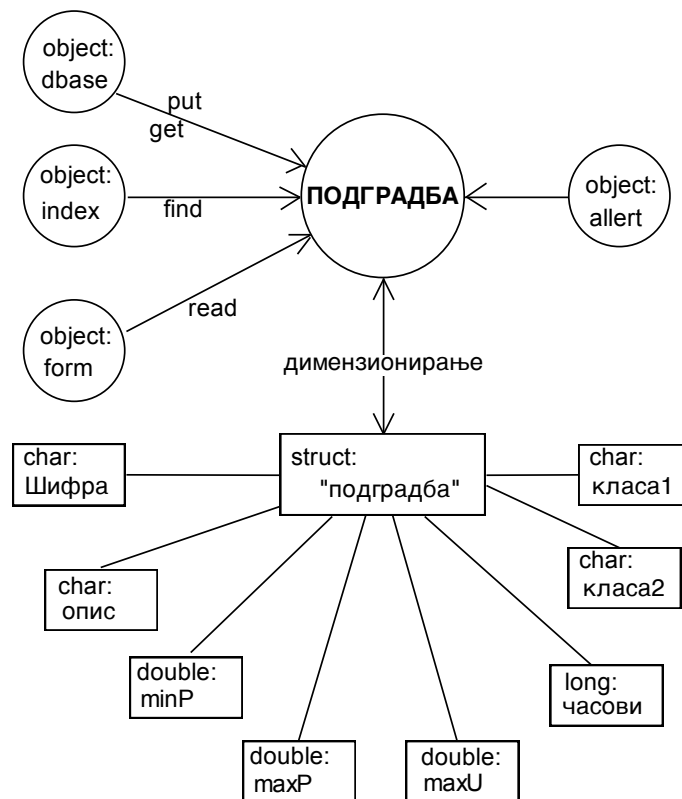
Операции кои се вршат врз објектите се:

- read - внесување на податоците;
- put - запишување на диск;
- get - читање од диск;
- find - барање на објектите од диск.



Овие операции се однесуваат за сите објекти и се вршат со посредство на горе споменатите библиотечни објекти. Поединечно за секој објект издвоени се операциите кои се бараат од објектите, а тоа се:

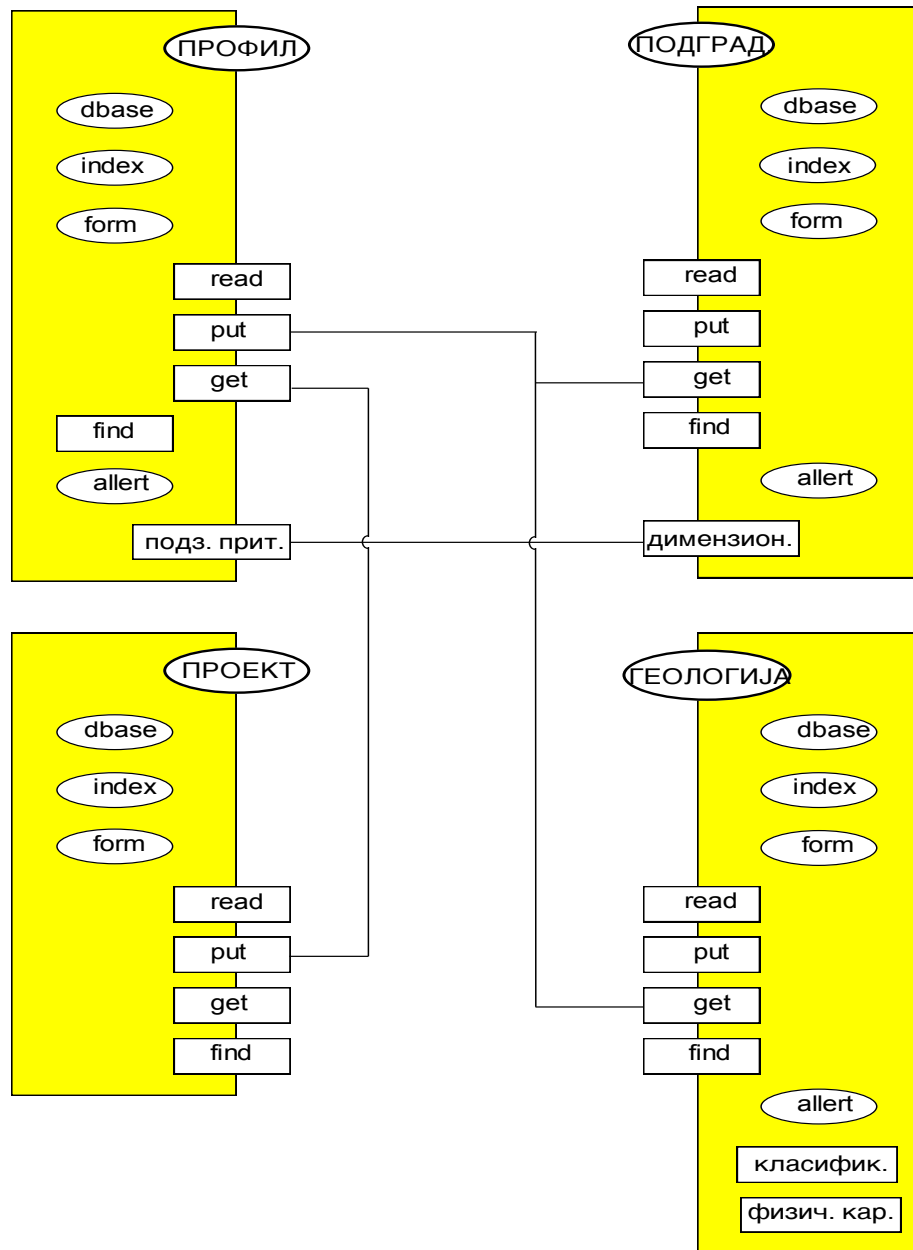
1. за објектот "геологија" :
  - класификација на геолошкиот материјал;
  - утврдување на меродавните физички карактеристики;
2. за објектот "подградба" :
  - димензионирање на подградбата;
3. за објектот "профил" :
  - утврдување на подземните притисоци;
4. за објектот "проект" :
  - не се бараат посебни операции.



Слика 4-11 Ајрибујии на објектиои "Подградба"

Во третиот чекор, утврдување на видливоста на секој објект во релација со другиите објекти, се идентификува статичката зависност помеѓу објектите и класите (кои податоци или операции се достапни и за други објекти). Целта е да се усвои топологијата на објектите од моделот на реалноста (сместувањето на податоците и функциите-

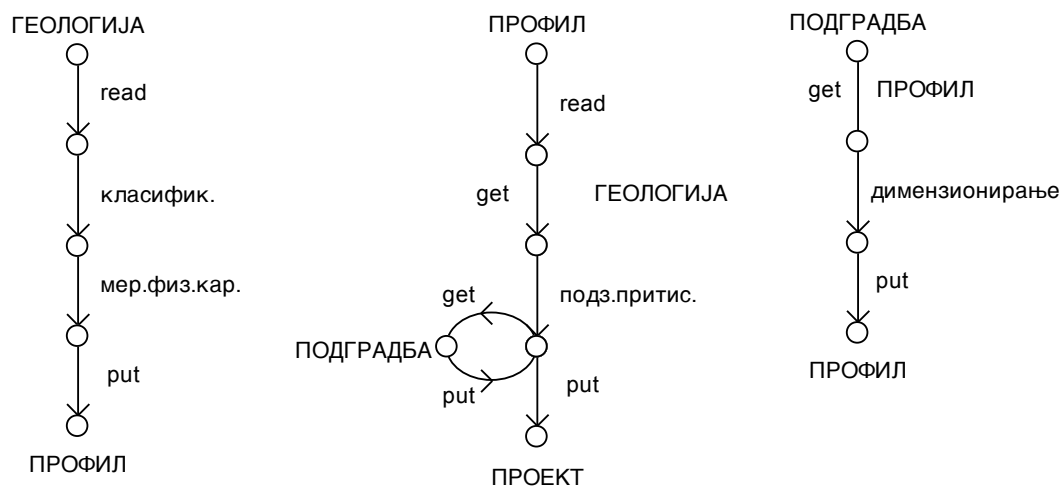
методите во објектот, гледан како една целина). Ова е претставено графички на сл.4-12 преку "пакети". Атрибутите затворени во "пакетот" не се видливи за другите објекти, додека атрибутите кои до пола се излезени можат да комуницираат и со надворешните објекти и со другите атрибути од самиот објект.



Слика 3.12 - Објекти на подсистемот "прекуирање" - видливост и интерфејси

На сл.4-12 исто така е претставен и интерфејсот за секој објект, што го претставува и наредниот чекор при утврдувањето на моделот.

Недостаток на овој начин на претставување на објектно-ориентиранот модел (сл.4-12) е што не се гледа постапноста т.е. динамиката на одвивањето на комуникациите помеѓу објектите со што не е задоволен вториот чекор од моделирањето. Поради тоа дадени се дијаграмите на сл.4-13 во која е претставен животниот циклусот на објектите.

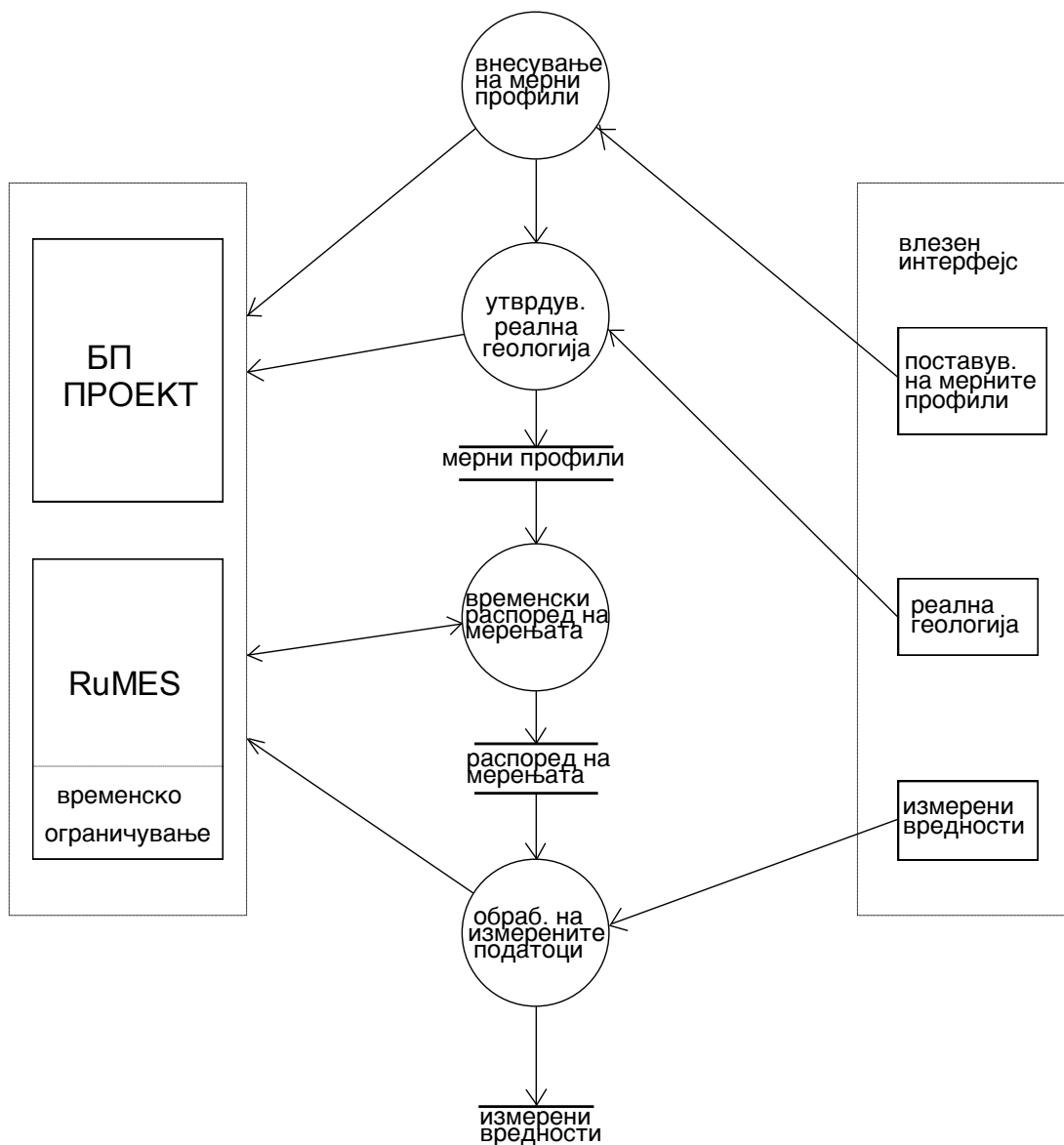


Слика 4-13 - Животен циклус на објектите

## 4.5. ИНФОРМАЦИОНЕН ПОДСИСТЕМ "ГРАДБА"

### 4.5.1. АНАЛИЗА НА ПОДСИСТЕМОТ

Токот на информациите за подсистемот "градба" е даден преку DFD на слика 4-14. Подсистемот глобално се состои од три дела: влезен интерфејс, активна база на податоци и дел во кој се врши обработката на податоците.



Слика 4-14. Информационен подсистем "Градба"

Влезниот интерфејс се состои од модули кои овозможуваат внесување на податоците за мерните профили, реалната геологија и измерените вредности на конвергенциите или притисоците. Внесувањето на податоците е интерактивно, додека мерните вредности може да се внесуваат и автоматски, доколку инструментите се опремени со сензори. Податоците кои вака влегуваат во системот постапно се внесуваат во базата на податоци. Најпрво се утврдува мерниот профил, односно стационожата на профилот каде што се поставуваат мерните инструменти. Во овој момент RuMES генерира услов во кој се заведени мерниот профил и проектираната геологија. Условот се активира во следниот чекор: се задава реалната геологија и се споредува со проектираната. Тука се можни три случаи:

1. Реалната геологија се совпаѓа со проектираната:
  - Условот е задоволен и се оди на наредниот чекор.
2. Реалната геологија не се совпаѓа со проектираната, но постои во прогнозниот геолошки профил (предвидена е во проектот):
  - Условот не е задоволен и се повикува подсистемот "проектирање", модулите за утврдување на подземниот притисок и димензионирање на подградбата (се врши ажурирање на проектот).
3. Реалната геологија не се совпаѓа со проектираната, и не постои во прогнозниот геолошки профил (не е предвидена во проектот):
  - Условот не е задоволен, се повикува системот "проект", и се внесуваат сите валидни податоци за реалната геологија и натаму се ажурира проектот: подземни притисоци и димензионирање.

Следниот чекор може да се изврши и без задоволување на претходниот услов во случаите 2 и 3, но со тоа што тогаш тој останува активен и RuMES алармира при секое активирање на системот. Во спротивно, кога ќе се ажурира проектот, условот се деактивира.

По внесувањето на податоците за мерниот профил се прави терминскиот распоредот за мерењата т.е. се утврдуваат наредните услови во кои се дадени временските ограничувања за одредениот мерен профил (се одредуваат времињата кога треба да се извршат потребните мерења - во интервали од неколку часа до десетина дена). Паралелно со временските ограничување во истите услови се задаваат и максималните вредности за конвергенциите (деформациите), односно, притисокот. Овие максимални вредности се утврдени врз

основа на проектираната подградба, која треба постапно да се вградува. При евалуирањето овие услови се деактивирани, а ги активира системскиот часовник непосредно пред да се изврши мерењето. При тоа можни се два случаи:

$$1. \quad U \leq \max U \quad \text{или} \quad P \leq \max P$$

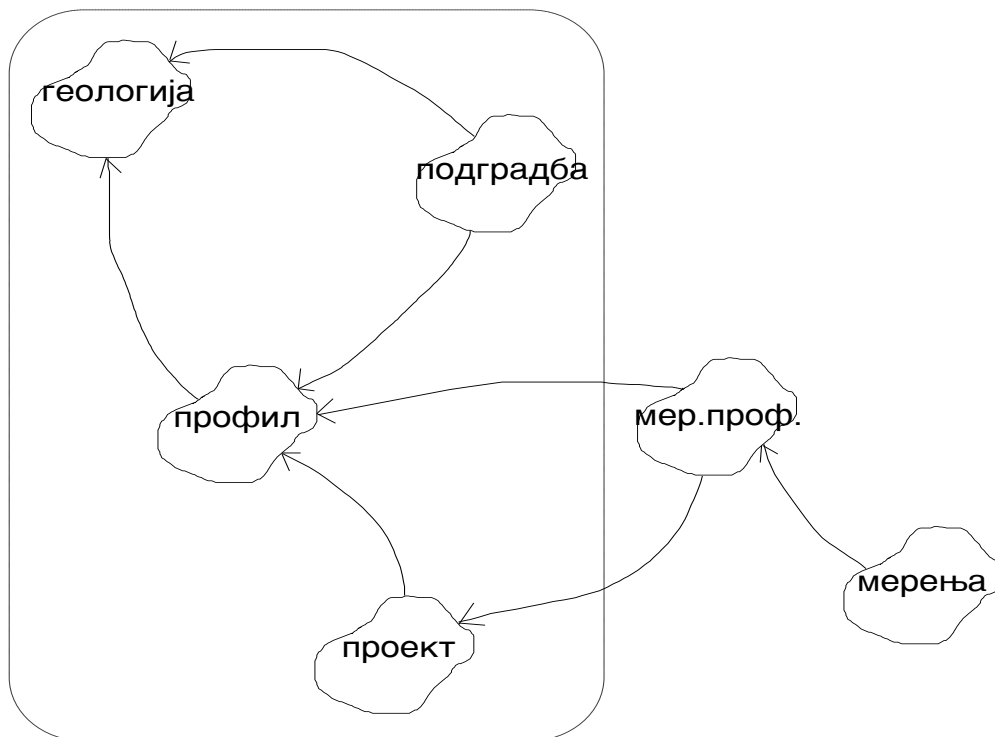
Условот е задоволен. Постапената подградба е во ред и не е потребно да се појачува.

$$2. \quad U > \max U \quad \text{или} \quad P > \max P$$

Условот не е задоволен. Потребно е појачување на подградбата.

Условите кои се генерираат во овој чекор се деактивираат по внесувањето на измерените вредности. Во случај да не е потврдено дека подградата е појачана условот останува активен и алармира при активирањето на системот.

На крај треба уште да се каже дека сите податоци што се внесуваат се складираат во базата на податоци за покасна употреба во подсистемот проектирање.



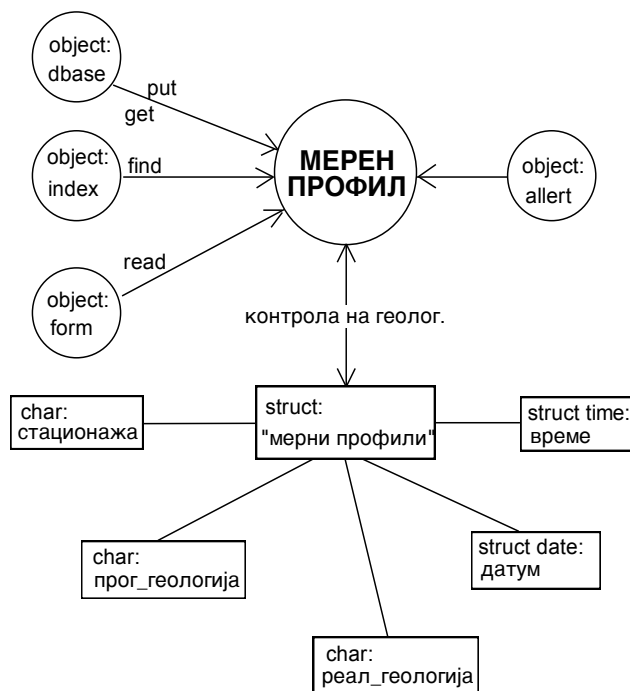
Слика 4-15. Објекти на подсистемот "Градба"

#### 4.5.2. ОБЈЕКТНО ОРИЕНТИРАН МОДЕЛ НА ИПС "ГРАДБА"

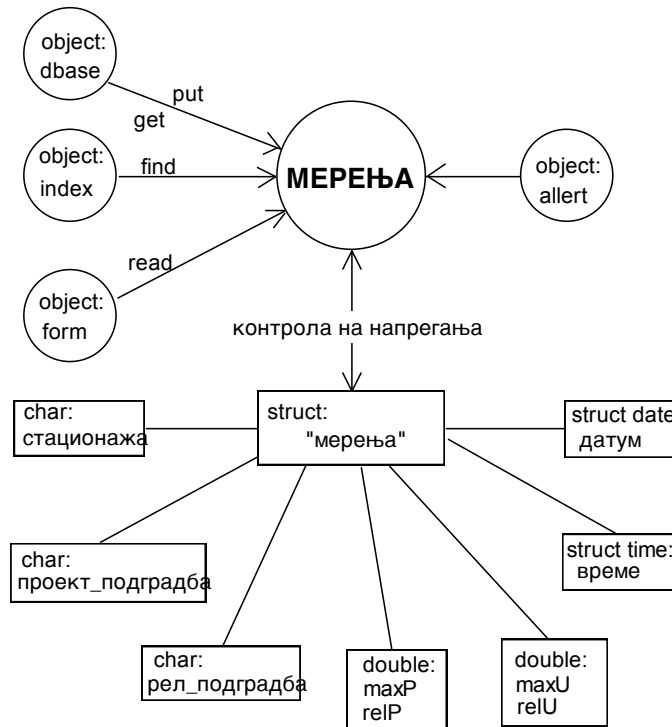
Основен ентитет кој се појавува во овој подсистем се мерењата. Мерењата се извршуваат на одредени мерни профили. Тоа се всушност другите објекти кои се појавуваат во системот. Освен нив во овој подсистем своја улога си имаат и објектите од подсистемот "проектирање" (сл. 4-15.), кои не се активно вклучени во овој подсистем, туку се повикуваат по потреба. Конкретно, тука се можни повеќе инстанци од објектот мерни профил за еден проект. Исто така постојат повеќе мерења за еден мерен профил.

Објектите треба да ги содржат следниве податоци:

- "мерни профили" - напречен профил на кој ќе се вршат мерењата - ги содржи стационажата на која се наоѓа, часот и датумот на избивање на профилот т.е. на поставување на мерниот профил, прогнозната геологија (земена од подсистемот "проект" во моментот кога се задава мерниот профил) и реалната геологија (Сл. 4-16.).
- "мерења" - податоци за извршените мерења на подземниот притисок или конвергенциите - содржи: стационаж на мерниот профил, час и датум на мерење, податоци за проектираната и вградената подградба (сл. 4-17.).



Слика 4-16. Атрибуции на објектот "Мерен профил"



Слока 4-17. - Атрибуции на објектите "Мерења"

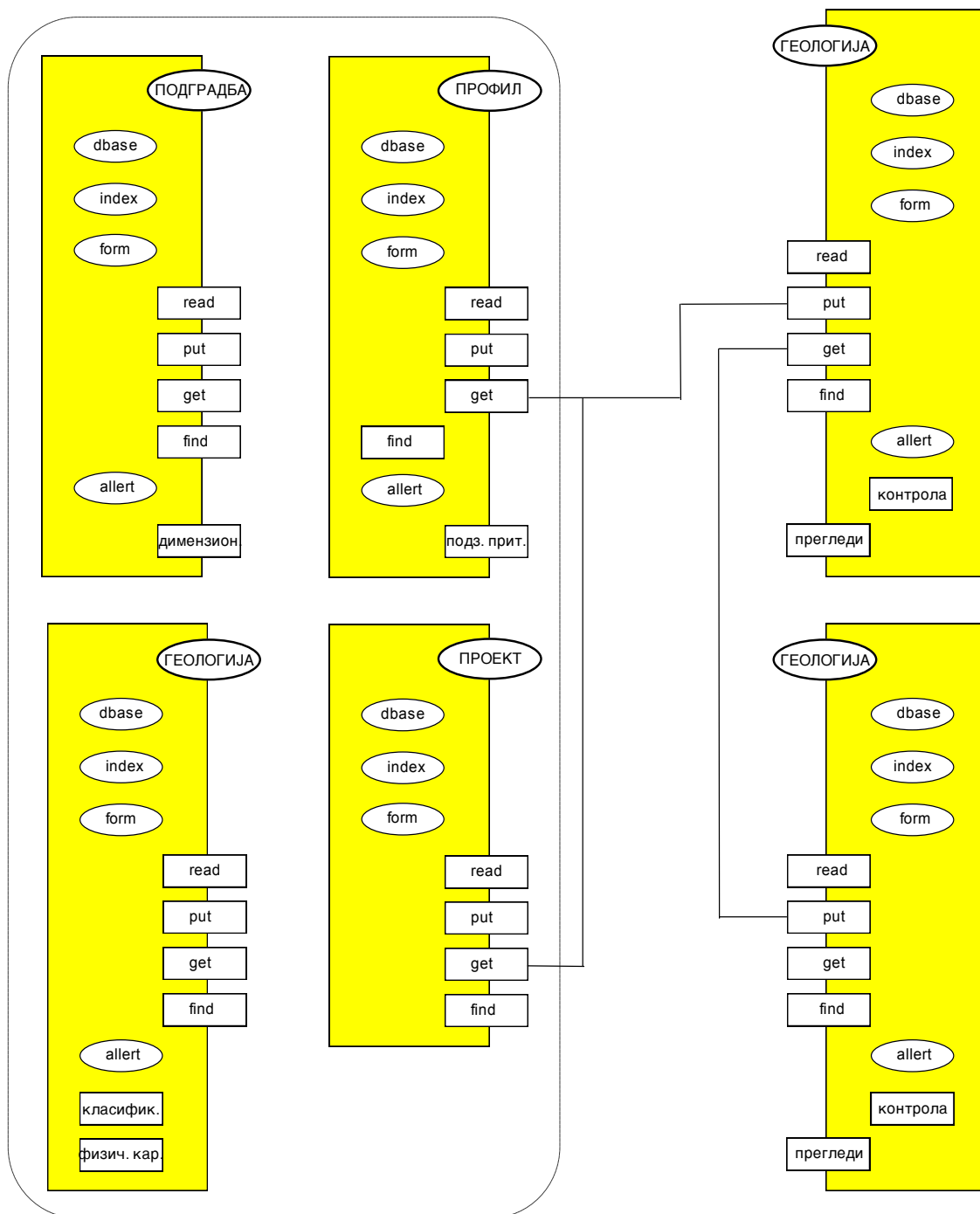
Исто како и во подсистемот "проект" и во овој подсистем како дел од атрибутите на гореспоменатите објекти се јавуваат и библиотечните објекти:

- database - овозможува запишување и читање од диск
- btree - овозможува индексирање и пребарување на датотеките.
- dialogue - прозор (форма, маска) кој служи како кориснички интерфејс за работа со податоците.
- alert - дел од корисничкиот интерфејс кој служи за информирање или опоменување на корисникот за однесувањето на системот во врска со одреден објект.

Освен операциите read, put, get, find кои се вршат врз сите објекти за објектот "мерен профил" карактеристична е операцијата која врши контрола на проектираната (прогнозираната) геологија со реалната која се задава при утврдувањето на мерниот профил. За објектот "мерења" карактеристична е операцијата која врши контрола на измерените вредности во однос на проектираните. За овие два објекта, исто така, потребно е да се предвиди и операција која по потреба ќе



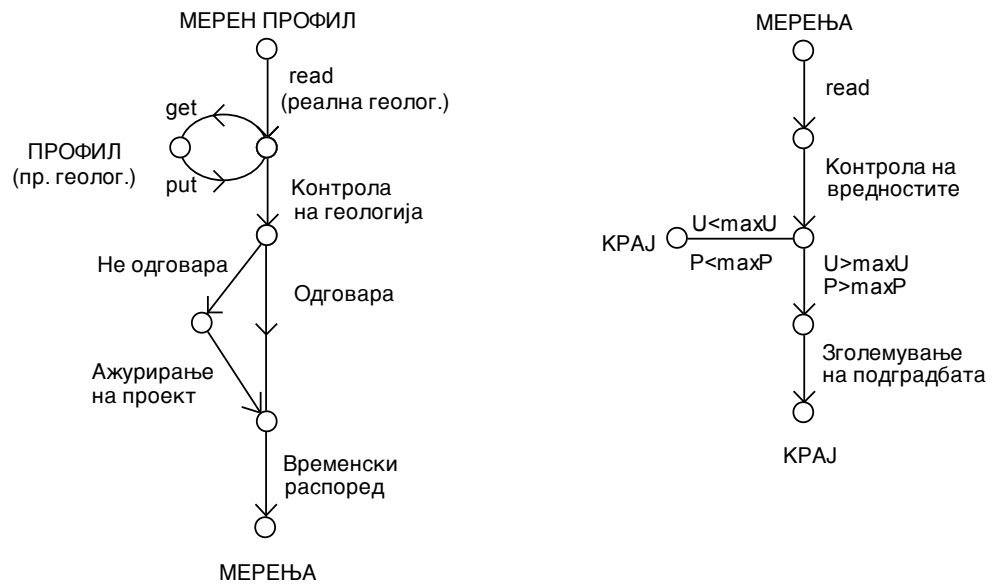
дава преглед на поставените мерни профили или ќе дава преглед на извршените мерења.



Слика 4-18. - Објекти на подсистемот "градба" - видливост и интерфејси

Интерфејсот на објектите, како и видливоста на секој објект претставени се на слика 4-18.

На слика 4-19. претставен е животниот циклус на објектите каде што прикажана постапноста во одвивањето на операциите.

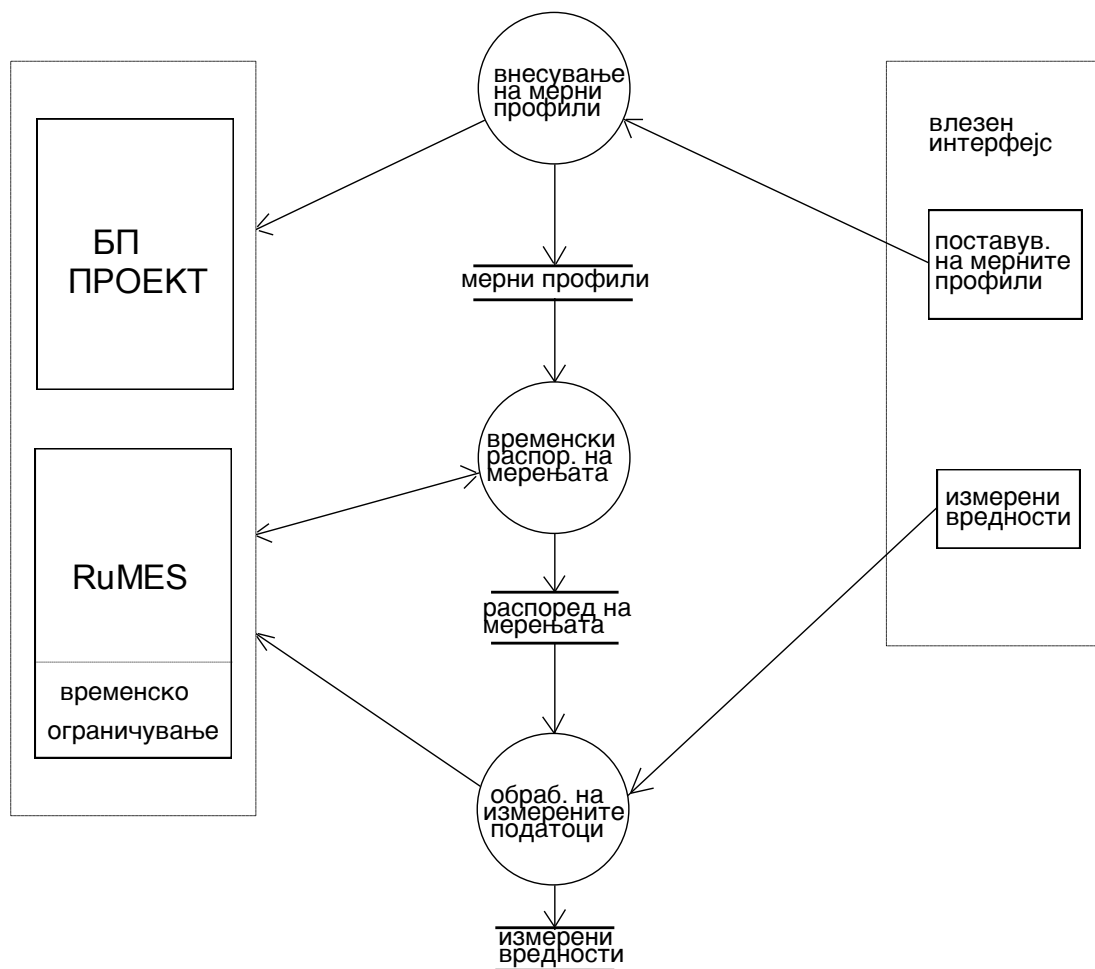


Слика 4-19. - Животен циклус на објектите од подсистемот "градба"

## 4.6. ИНФОРМАЦИОНЕН ПОДСИСТЕМ "ЕКСПЛОАТАЦИЈА"

### 4.6.1. АНАЛИЗА НА ПОДСИСТЕМОТ

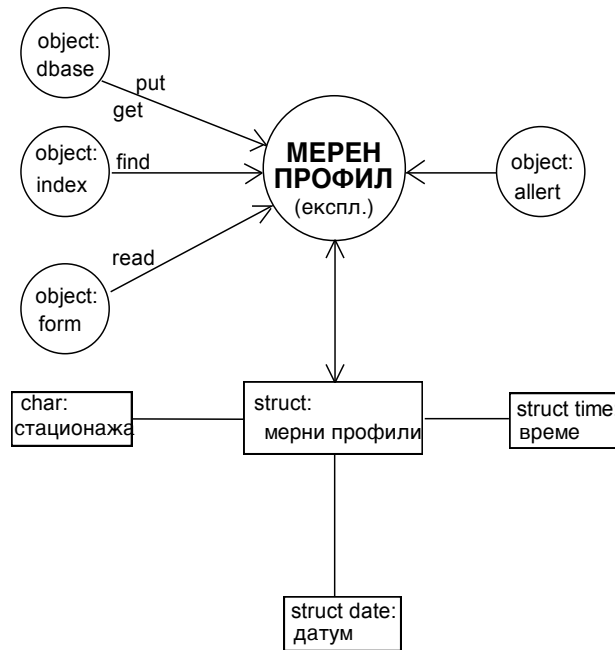
Сличен на подсистемот "градба" е и подсистемот "експлоатација" (сл. 4-20.). Исто и како претходниот подсистем и овој, глобално, се состои од три дела: влезен интерфејс, активна база на податоци и дел во кој се врши обработката на податоците.



Слика 4-20 Информационен подсистем "експлоатација"

За разлика од подсистемот "градба" во влезниот интерфејс не постои модулот за внесување податоци за реалната геологија. Соодветно на тоа и се прескокнува чекорот каде што се генерира

услов за проверка на геологијата. Сето друго што е кажано за подсистемот "градба" важи и за подсистемот "геологија". Овде само треба уште да се напомене дека интервалите помеѓу мерењата се од ред на големина неколку месеци.

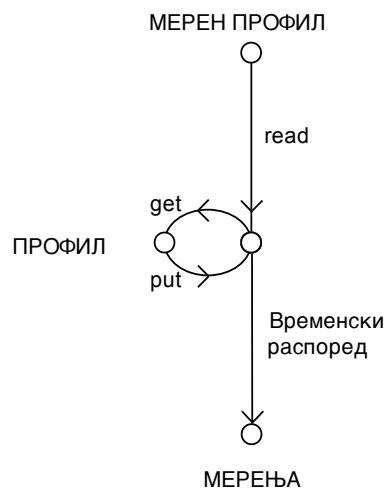


Слика 4-21 Атрибуции на објектот "Мерен профил" од ИПС "Експлоатација"

#### 4.6.2. ОБЈЕКТНО ОРИЕНТИРАН МОДЕЛ НА ИПС "ЕКСПЛОАТАЦИЈА"

Како што веќе рековме, подсистемот "експлоатација" е сличен на подсистемот "градба". Идентифицирани се истите објекти како во ИПС "градба", но со минимални разлики за објектот "мерен профил": не се задаваат податоците за реалната и проектираната геологија. Значи "мерен профил" ги содржи само податоците за стациоњата и часот и денот на поставувањето на профилот (од кога ќе се следи однесувањето на тунелот во тој профил). Овој објект е прикажан на слика 4-21. Се што е кажано порано за другиот објект "мерења" важи и за овој подсистем.

При идентификацијата на операциите за овие објекти утврдено е дека се идентични со операциите за истите објекти од подсистемот градба само со единствена разлика што за објектот "мерен профил" не постои операцијата "контрола на геологијата". Според тоа разлика за овој објект ќе се појави и во дијаграмот на кој е претставен животниот циклус на објектот (сл. 4-22).



Слика 4-22 - Животен циклус на објектот "мерен профил" од подсистемот "експлоатација"

## 4.7. ПОДСИСТЕМ ЗА УПРАВУВАЊЕ СО АКТИВНАТА БАЗА НА ПОДАТОЦИ

Многу апликации (како на пример компјутерски подржаното производство, контролата на процеси или управувањето со мрежи) бараат пристап кон големи бази на податоци и бази на знаење, и бараат навремен одговор во критичните ситуации. За таквите апликации најважно е да се следат условите дефинирани во базата на податоци, и кога тие услови ќе се исполнат да повикува одредени акции, кои се предмет на соодветни временски ограничувања [11].

Таквите временски ограничени апликации не се добро опслужени од страна на конвенционалните системи за управување со бази на податоци (DBMS). Тоа е поради тоа што ваквите DBMS се пасивни т.е. тие вршат контрола на условите, или вршат трансакции, единствено кога тоа се бара експлицитно од страна на корисникот или програмот.

Еден од приодите за активирање на пасивните бази на податоци е да се искористи можноста програмот да врши периодични прозивки (периодично поставување на прашања) на базата на податоци, при што се контролира дали ситуацијата т.е. условот кој се набљудува се исполнил [9],[11],[17]. За да бидеме сигурни во правовремената детекција на ситуациите и за да се одговори на дадената ситуација, програмот мора да врши чести прозивки. Во овој случај постои опасност да се појават многу празни одговори.

Алтернатива на прозивките е секој програм кој ја ажурира базата на податоци да ја проверува ситуацијата по извршеното ажурирање преку вметнување на посебни процедури [9],[11],[17]. Меѓутоа и тука постои проблем, во смисла на тоа дека со ваквото вметнување на процедури се намалува модуларноста на програмот т.е. се отежнува одржувањето на програмот: било каква измена на ситуацијата која се набљудува или на акциите кои одговараат на таа ситуација ќе бара модифицирање на секој програм кој ја ажурира базата.

Системот за управување со активни бази на податоци се обидува да ги осигури и модуларноста и можноста за временско реагирање.

Ситуациите, акциите и временските ограничувања се одредени декларативно во системот. Сега системот ги следи ситуациите и ги активира соодветните акции ако се исполни условот [17].

Значи за разлика од традиционалните системи за управување со бази на податоци, кои се пасивни и кои реагираат само на барање на корисникот или апликацијата, активните бази на податоци овозможуваат да се добие правремена реакција на специфираните критични ситуации. Ова посебно доаѓа до израз кај временски ограничените апликации - што е случај и со нашиот систем. Тука станува збор за подсистем кој е заеднички за претходните подсистеми. Се состои од класична база на податоци и механизам за активирање на самата база.

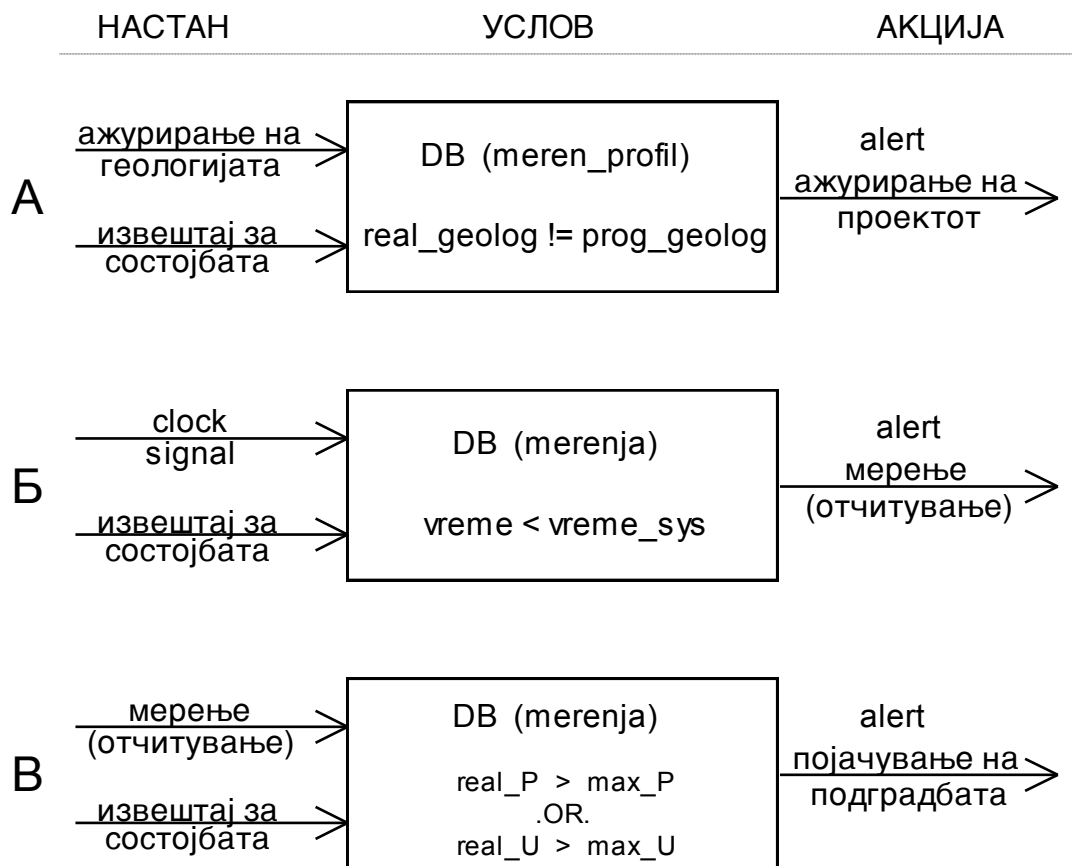
Самиот подсистем се базира на ЕСА (event-condition-action = настан-услов-акција) правила. При појава на настан се проверува условот и ако е исполнет се извршува акцијата. Подсистемот има можности да генерира (додава) или брише правила, како и да врши нивно активирање или деактивирање - во зависност од потребите [9].

Правилата се претставени во облици како на слика 4-23. Тука се дадени три примери.

Под "А" како настан се појавува моментот кога се врши отварање на нов профил (мерен профил). Со увид на лице место се утврдува видот на геолошкиот материјал. Ако материјалот (геологијата) не одговара на прогнозираниот тогаш се врши ажурирање на проектот т.е. се бара тип на подградба која одговара на тој геолошки материјал. По тоа се додаваат во базата правилата вградени преку објектите "мерења", во кои се задаваат термините (времињата) кога треба да се изврши мерењето, како и максималните вредности кои се дозволени за тој тип на подградба.

Правилото под "Б" го прикажува случајот на прозивка од страна на системот кога се споредува дали е дојден моментот за извршување на мерењето. Истото се проверува и во случај да се побара, на пример, распоредот за мерење. Од градежна гледна точка, многу е битно да се извршат мерењата според однапред утврдените термини (посебно почетните мерења), поради што треба да се потенцира ова правило.

Друг облик на правило е кога се споредуваат измерените вредности. Во случај отчитаните вредности да се поголеми од максимално дозволените тогаш системот треба да алармира и да бара вградување на појака подградба. Тука е особено битна правремената реакција поради опасноста од хаварија (обрушување) во тунелот.



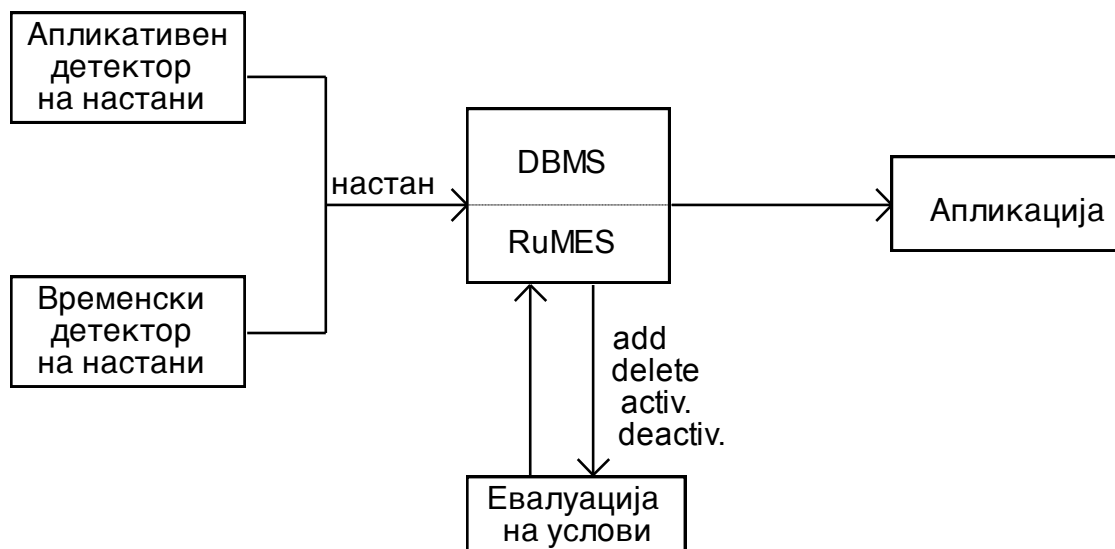
Слика 4-23. - Облик на вградениите правила

Како што веќе нагласивме состојбите т.е. правилата, акциите и временските ограничувања се вградени во подсистемите. На тој начин се надгледува ситуацијата во базата на податоци и по потреба се активираат акции, односно се испраќаат предупредувања до корисникот. Функционирањето на подсистемот се одвива како на слика 4-24.

Самите правила всушност се претставени преку објекти, во конкретниот случај објектите "мерен профил" и "мерења", кои претходно веќе се претставени во подсистемите "Градба" и "Експлоатација". Активирањето на базата е извршено преку



вметнувања и преку прозивки. Тоа всушност е еден начин на приближување на пасивната база на податоци кон временски ограничена апликација.



Сл. 4.24. Подсистем за управување со активната база на податоци

На крајот тука уште еднаш треба да напоменеме дека правилата се генерираат на одреден сигнал од апликацијата, а исто така на сигнал се проверува условот или се деактивираат правилата ако веќе не се валидни.

## 5. ИМПЛЕМЕНТАЦИЈА НА СИСТЕМОТ

Прототип на системот е имплементиран во C++.

При изборот на програмскиот јазик немаше многу дилеми. За развој на ваков комплексен систем е потребен ефикасен и флексибилен јазик и кој, нормално, подржува објектно ориентирано програмирање.

C++ во последните неколку години е еден од најчесто повикуваните програмски јазици, кој сам по себе се наметнува со своите одлики. Определбата прототипот на системот да се имплементира во C++ е поткрепена со неколку елементи:

- Најпрво и најбитно, C++ е јазик за објектно ориентирано програмирање.
- Второ, C++ е изведен од C. Со неколку исклучоци тој претставува надградба на C, што значи дека се што е применливо во C, е применливо и во C++. Тука треба да се акцентира моќта на C, како современ програмски јазик, и можноста за примена на огромниот број библиотечни кодови за C.
- Друг елемент кој не е за потценување е и преносливоста на кодот на разни машини и платформи.
- И на крај овде мора да ги споменеме и библиотеките на класи, кои поради познатите предности на објектно ориентираното програмирање се уште пофлексибилни и многу лесно применливи.

При имплементацијата е користена следната постапка:

- најпрво се креирани класите (според утврдениот модел),
- потоа се креирани објекти (инстанци) за соодветните класи и
- на крај се имплементирани пораките за комуникација помеѓу објектите - пораките се еквивалент на повикот на методите (функциите членки) од страна на соодветните објекти.

Отежнувачка околност при кодирањето беше комплексноста на системот. Овде се мисли во смисла на утврдување на границите во кои ќе се оди со програмирањето. Постоеше опасност да тие граници бидат премногу широки па имплементацијата на прототипот да одземе многу време. Во тој случај, исто така под знак на прашање ќе дојде и валидноста на системот, во смисла на коректност на подеталните решенија, како во информатичка - тако и во градежна смисла. Исто така постоеше опасност и да тие граници бидат премногу тесни со што функционалноста на системот, исто така, би била доведена во прашање.

Глобално гледано прототипот на системот ги содржи сите проектирани подсистеми. За секој подсистем, различно, направени се одредени апроксимации со што не се губи функционалноста на системот како целина.

За подсистемот "Проектирање" имплементиран е влезниот интерфејс, додека модулите за геолошка класификација и димензионирање на подградбата се симулирани. Самите овие модули би требало накнадно да бидат посебно разработени. Излезните резултати од овој подсистем, поради симулацијата, се внесуваат мануелно во базата на податоци. Подсистемите "Градба" и "Проектирање" имплементирани се така да работат само за наједноставни случаи. Исто така и овде се усвоени одредени симулации кои ја упростуваат работата на прототипот.

Надградбата на овие подсистеми треба да се изведе со наследување на објектите, при што ќе се наследува само интерфејсот и најосновните методи.

Нормално, при имплементација на системот, најпогодно би било да се користи некое проширување на C++ за управување со објектно-ориентирани бази на податоци. Но во недостаток на тоа проширување, системот за управување со базата на податоци е имплементиран како еден вид на релациона база, што при имплементацијата на комерцијален систем не мора да се задржи. Тука е искористена една едноставна библиотека на класи која овозможува читање и пишување во полиња, како и движење низ тие записи. За директен пристап кон записите постои и класа за индексирање која работи како Б-дрво. Активирањето на базата на податоци е извршено преку вметнување и

преку прозивки. За евентуалениот комерцијален систем овој дел би требало да се разработи подетално.

Посебно внимание е посветено на креирањето на корисничкиот интерфејс. Тука се појавија три моменти кои треба да се потенцираат:

- уредување на менијата (т.е. начинот на пристап кон поодделни сегменти од системот),
- начинот на внесување на податоците и
- презентирање на пораките, предупредувањата, кои системот ги упатува до корисникот.

Менијата се креирани врз принципот на мени и подмени. Основното мени е постојано присутно на екранот во хоризонтална линија (најгорниот ред од екранот), додека подменијата се појавуваат како паѓачки (pull down).

Внесувањето на податоците се врши преку форми (маски), кои освен што имаат можност за впишување на потребните податоци, туку имаат и можност за избор на неколку понудени варијанти од податоците кои треба да се внесуваат. Освен тоа во самите форми се појавува и посебно мени кое овозможува движење низ записите (рикордите) на базата на податоци, корегирање на одделните записи, како и бришење на означените записи.

Пораките кои се појавуваат до корисникот се појавуваат во вид на прозори преку работниот екран. Корисникот мора да потврди дека ја прочитал пораката.

Заради поголема ефикасност при кодирањето се користени библиотечни класи. Конкретно, класите "dBase" и "BTree" се користени при имплементацијата на функциите членки кои се користат за работа со базата на податоци. Понатаму библиотеката "Screen" е користена за имплементирање на корисничкиот интерфејс, односно менијата, формите за упис и пораките.

Прототипот на системот е развиен на IBM PC AT компатибилни компјутери и со програмскиот пакет TURBO C++ v.1.01 за DOS. На ова ниво на развој на системот овие ресурси во целост ги задоволуваат потребите. Меѓутоа при евентуалниот развој на комерцијална верзија би требало да се избере појака машина со изразени графички можности и соодветен компајлер.

Времето потребно за кодирање на прототипот ефективно изнесуваше околу два месеци, но во тие два месеци не е влезено времето потребно за проучување на библиотеките на класи кои се употребени. Кодирањето е извршено во два наврати т.е. со две итерации, но се појавува простор и за понатамошна разработка на системот. Според досегашните сознанија за добивање на комерцијална верзија, која би била со многу поголеми можности, потребна е неколку годишна работа на тим кој би се состоел од стручњаци во областа (тунелоградба) како и од стручњаци информатичари. Прототипот е тестиран на реален пример од веќе изградениот опточен тунел на хидросистемот "Лисиче", а искористени се мерењата на конвергенциите во тунелот кои ги извршив во соработка со "Геомаврово" - Скопје, во периодот од април 1992 до септември 1992. Системот на ова ниво на развој сосема ги задоволи очекувањата.

## 6. ЗАКЛУЧОК

За да се обезбеди продуктивност во работата на инженерскиот кадар во тунелоградбата, како и да се добијат рационални и економични објекти - неопходен е развој на информациона систем кој ќе помага при проектирањето и градбата и ќе ја следи експлоатацијата на тунелите.

Поради специфичноста на тунелите, како градежни објекти, проектирањето и градбата во голема мера се базираат на искуството. Како начин да се искористи искуството од претходно изградените тунели, се јавува потреба од собирање и систематизација на податоците т.е. искуството од тие веќе изградени тунели.

Повеќе автори имаат работено на различни системи од областа на тунелоградбата. Заеднички одлики на овие системи е што нудат ограничени решенија. Овде се мисли на ограничување по фази (на пример само за проектирање) или пак за повеќе фази, но за одредена методологија на работа (на пример за пробивање на тунели со замрзнување на земјата) или само за одреден тип на тунели (на пример само за плитки тунели). Друга заедничка одлика на овие системи е што во најголем број се развивани по класична методологија, така да нивната евентуална надградба е доведена во прашање.

За разлика од класичното, процедурално, програмирање каде што при решавањето на проблемите се сконцентрираме на постапката т.е. алгоритмот за решавање, кај објектно ориентираното програмирање работата се насочува кон развој на модел на реалниот свет т.е. на утврдување на интеракциите помеѓу објектите кои го сочинуваат тој модел. Со помош на своите клучни концепти "апстракцијата", "енкапсулацијата" и "хиерархијата на класи", објектно ориентираното програмирање е посебно погодно за развој на големи и комплексни информациона системи.

Размислувањата при ваквиот пристап се свртени кон проблемот кој ќе се решава, а не кон начинот на кој тој проблем ќе се кодира т.е. имплементира. Поради тоа, објектно ориентираните програми се одликуваат со голема флексибилност.

Постапката за развој на објектно ориентираните програми се состои во неколку генерални чекори:

- идентификација на класите (објектите);
- доделување на атрибути и методи на класите;
- утврдување на врските помеѓу класите и
- уредување на класите во хиерархии.

За развојот на моделот на информациониот систем е извршено прилагодување на оваа генерална постапка, посебно за решавање на ваков тип на проблеми. Најпрво прецизно е дефиниран проблемот на проектирање, градба и експлоатација на тунелите, при што се идентификувани глобалните задачи на системот. Понатаму е извршена глобална анализа на системот и поделба на подсистеми, кои се претставени шематски. Секој од овие подсистеми, понатаму, е претставен преку DFD (дијаграм на ток на податоците), од каде се идентификувани објектите за секој подсистем. Атрибутите за секој објект се дадени шематски, а нивните врски се претставени преку "пакети".

Постапката се покажа многу ефикасна и флексибилна, бидејќи овозможува итеративно враќање на претходните чекори, со цел да се добие модел кој ќе биде задоволителна копија на реалниот свет.

Како недостаток на оваа постапка се јавува ограниченоста при претставувањето на хиерархиите на класи, но во конкретниот случај тој недостаток не влијаеше многу на крајниот резултат. Секако дека системот го користи концептот за хиерархија на класите. Едноставно, овде потенцијалните основни класи, класи од кои би се изведувале други, се разгледувани скоро на ниво на апстрактни класи. Условно тоа значи дека изведените класи ќе го наследуваат само интерфејсот и само неколку методи кои би требало да се исти за сите наредни - изведени класи.

На крај е имплементиран прототип на системот во C++ и е извршено тестирање на реален пример. Во системот е вградена активна база на податоци каде што правилата се претставени како објекти, а работат по ECA (event-condition-action) принципот.

.....

Со овој модел е поставена основа за развој на многу комплексен систем. За понатамошен развој на системот е потребно само да се имплементираат неопходните нови објекти, кои ќе ги превземаат карактеристиките на претходно имплементираните објекти. Со ваквото изведување на нови објекти од основните, веќе имплементирани, тоа се постигнува максимална адаптивност на системот.

Развојот на комерцијален систем би требало да го реализира тим на експерти од доменот (тунелоградбата) и стручњаци - информатичари. Реализацијата на ваков систем би траела неколку години.



## ЛИТЕРАТУРА

## ЛИТЕРАТУРА

1. ADACHI, K.; YASHIMA, K.; KOJIMA, K., "Developments in Geotechnical Aspects of Embankments, Excavations, and Buried Structures", AABalkema, Rotterdam, Netherlands, pg. 291-329, 1991
2. BATTELIND, D., "Computer design of tunnelling with soil freezing", Acta geotechnica 1990, pg. 1441-1445
3. BIKINEEV, M. G., "Computer-aided design (CAD) system 'Tonel'", Hydro-technical Construction (English translation of "Гидротехническое Строительство") v. 25 n 5 Nov 1991 pg. 249-255
4. BIKINEEV, M. G., "'Tunnel' and computer-aided design system", Гидротехническое Строительство, н 5 Мај 1991 гг. 11-15
5. BORLAND TURBO C++ v1.01, "Programer's Guide", 1992
6. BOOCH, G., "Object-Oriented Development", IEEE Transaction on software engineering, Feb 1986
7. BULYCHEV, N. S.; ROZENVASSER, G. R.; SHAMRIN, Yu. E., "Analysis of prefabricated panels of collection tunnels in undermining soil massive", Основанија, Фундаменти и Механика Грунтов n 4 Jul-Aug 1991  
8. pg. 19.-21
9. ШАКРАВАРТИ, S., "Rule Manegement and Evaluation: An Active DBMS Perspective", SIGMOD RECORD, vol 18, No.3, Sept. 1989
10. ДАВЧЕВ, Д., "Структури на податоци, организација на датотеки и Бази на податоци", учебник, Универзитет "Кирил и Методиј", Скопје 1991
11. DAYAL, U., "Active Database Management Systems", 3rd International Conference on Data and Knowledge Bases, Jerusalem, June 1988
12. ISHII, M.; ISHIMURA, K.; NAKAYAMA, T., "Management and application of geotechnical data. The geotechnical data information system of the Tokyo metropolitan government", Environmental Geology and Water Sciences v.19 n3 May-Jun 1992, pg. 169-178

13. KAKOTO, T.; SKIBNEWSKI, M. , "Engineering decision support of automated shield tunnelling", Journal of Construction Engineering and Management v117 n4 Dec 1991 pg. 674-690
14. LEE, C.; STERLING, R., "Identifying probable failure modes for openings using a neural network", International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts v 29 n 1 Jan 1992 pg. 49-67
15. MICROSOFT C/C++ v7.0, "C++ Tutorial", 1992
16. POPOVIC, B., "Tuneli", Gradevinska knjiga, Beograd 1987
17. ROSENTHAL, A., "Situation Monitoring for Active Databases", Proceedings of the Fifteenth International Conference on very Large Data Bases, Amsterdam, 1989
18. SCHILDT, H., "Using Turbo C++", Osborne McGraw-Hill, 1990
19. SONG, D.; JUN, S. , "Non-linear rheological characteristics of rock like material and its mechanical effects on tunnel-surrounding-rock mass and support", Proceedings of the Asian Pacific Conference on Computational Mechanics, AABalkema, Rotterdam, Netherlands, pg. 801-806, 1991
20. VOSS, G., CHUI, P., "Turbo C++ Disk Tutor", Osborne McGraw-Hill, 1991
21. WIERINGA, J. R. , "Object-oriented Analysis, Structured Analysis and Jackson System Development", Object Oriented Approach in Information Systems, Elsevier Science Publishers B.V. (North Holland) 1991 IFIP
22. ЗАФИРОВА, М., "Нов Австриски Метод за градење на тунели", Годишен зборник на градежниот факултет - Скопје, 1985
23. ЗАФИРОВА, М., "За димензионирањето на надворешната тунелска цевка според Новиот Австриски Метод", Годишен зборник на градежниот факултет - Скопје, 1986
24. ЗДРАВЕВ, З., "Објектно ориентиран модел на информациона систем за проектирање, градба и експлоатација на тунели", ЕТАИ 1993, Охрид, септември 1993
25. ЗДРАВЕВ, З., "Примена на програмскиот пакет " TURBO C++ " при имплементација на програм за избор на оптимална метода на градба на тунели", семинарска работа, ЕТФ - Скопје, април 1992

# ПРИЛОГ

(тест пример)

## ПРИЛОГ

( тест пример )

Примерот на кој тестиран прототипот на системот се однесува на опточниот тунел од хидросистемот "Лисиче" кај Т. Велес. Тунелот е проектиран во текот на 1991 година. Изградбата на тунелот е завршена во почетокот на 1993 година.

Должината на тунелот изнесува  $L=508$  метри, а слободниот напречен профил е кружен со  $D=4,50$  метри. Тунелот според геолошката прогноза се требаше да се пробива низ три вида на геолошки материјал, кои се дадени подолу. Во надолжниот профил тунелот е претставен со повеќе десетици напречни профили, но за тестирањето избрани се пет карактеристични профили. Од овие пет профили градбата е следена на два профили и вредностите кои се измерени се дадени во примерот.

### ВЛЕЗНИ ПОДАТОЦИ ЗА ПОДСИСТЕМОТ "ПРОЕКТИРАЊЕ":

#### ПРОГНОЗНИ ГЕОЛОШКИ ПОДАТОЦИ:

001	Компактни гнајсеви, ушкрилени со кварцни жили и сочива,
	$\gamma = 27,0 \text{ KN/m}^2$ $\mu = 0,290$
	$E = 4000 \text{ MPa}$ $D = 2000 \text{ MPa}$
002	Ушкрилени микашести, графитични со кварцни жили
	$\gamma = 22,1 \text{ KN/m}^2$ $\mu = 0,350$
	$E = 3000 \text{ MPa}$ $D = 1300 \text{ MPa}$
003	Мермери, плочасти за банковити
	$\gamma = 27,0 \text{ KN/m}^2$ $\mu = 0,290$
	$E = 4000 \text{ MPa}$ $D = 2000 \text{ MPa}$

### НАПРЕЧНИ ПРОФИЛИ:

профил	стационажа	надслој Н	геологија
001	0+026.23	22.50 m	001
002	0+080.19	37.50 m	002
003	0+150.00	55.00 m	002
004	0+170.00	54.00 m	002
005	0+220.00	47.50 m	003

### ТИПОВИ НА ПОДГРАДБА :

(добиеени од модулот за димензионирање - симулација)

тип	опис на подградбата	geol.	maxU
T1	Шалуван армиран бетон d = 35cm, MB30	001, 003	15m m
T2	Армиран прскан бетон d = 5cm, MB30	002	20m m
	Шалуван армиран бетон d = 35 cm, MB30		
T3	Армиран прскан бетон d = 10 cm, MB30	002	30m m
	Анкери според измерени конвергенции		
	Шалуван армиран бетон d = 35 cm, MB30		

**РАСПОРЕД НА ТИПОВИТЕ ПОДГРАДБИ ПО  
НАПРЕЧНИ ПРОФИЛИ**

профил	стационажа	тип на подградба
001	0+026.23	T1
002	0+080.19	T2
003	0+150.00	T2, T3
004	0+170.00	T2
005	0+220.00	T1

**ВЛЕЗНИ ПОДАТОЦИ ЗА ПОДСИСТЕМОТ  
"ГРАДБА":**

**РЕАЛНИ ГЕОЛОШКИ ПОДАТОЦИ:**

При пробивањето на тунелот утврдено е дека реалната геологијата литолошки одговара на прогнозната геологија, но е со многу послаби физичко механички карактеристики. Поради тоа се воведени нови описи на геологијата за типовите 001 и 002 со ознака 004 и 005 (точните физичко механички карактеристики не беа познати) :

004	Испукани гнајсеви, ушкрилени со кварцни жили и сочива,
005	Распаднати ушкрилени микашисти, графитични со кварцни жили

РЕАЛЕН РАСПОРЕД НА ГЕОЛОГИЈАТА ПО  
НАПРЕЧНИ ПРОФИЛИ :

профил	стационажа	надслој Н	геологија
001	0+026.23	22.50 m	004
002	0+080.19	37.50 m	002
003	0+150.00	55.00 m	005
004	0+170.00	54.00 m	005
005	0+220.00	47.50 m	002

НОВО ПРОЕКТИРАНИ ТИПОВИ НА ПОДГРАДБА  
(согласно реалната геологија)

тип	опис на подградбата	geol.	maxU
T4a	Ременати $\Omega 21$ /1m	004, 005	32m m
	Армиран прскан бетон d = 12cm, MB30		
	Шалуван армиран бетон d = 35 cm, MB30		
T4б	Ременати $\Omega 21$ /1m	004, 005	35m m
	Армиран прскан бетон d = 12 cm, MB30		
	Анкери според измерени конвергенции		
	Шалуван армиран бетон d = 35 cm, MB30		



## РЕАЛЕН РАСПОРЕД НА ТИПОВИТЕ ПОДГРАДБИ ПО НАПРЕЧНИ ПРОФИЛИ

Со задавањето на реалната геологија системот автоматски врши распоредување на типовите подградби според реалната геологија.

профил	стационажа	тип на подградба
001	0+026.23	T4a
002	0+080.19	T3
003	0+150.00	T4a
004	0+170.00	T4a
005	0+220.00	T3

### ПОСТАВЕНИ МЕРНИ ПРОФИЛИ :

	профил	стационажа	датум	час
001	003	0+150.00	29/05/1992	10:00
002	004	0+170.00	23/06/93	02:30

**ИЗМЕРЕНИ ВРЕДНОСТИ :**  
(мерен профил 001, стационача 0+150,00)

ред.бр.	датум	час	конвергенција
1	29/05/93	22:00	0.00
2	30/05/93	15:30	0.30
3	02/06/93	11:00	7.03
4	24/06/93	08:00	12.25
5	30/06/93	14:00	12.43
6	07/07/93	22:30	13:79
7	23/07/93	13:00	13:78

**ИЗМЕРЕНИ ВРЕДНОСТИ :**  
(мерен профил 002, стационача 0+170,00)

ред.бр.	датум	час	конвергенција
1	24/06/93	08:00	0.00
2	24/06/93	13:00	0.72
3	25/06/93	12:00	1.01
4	30/06/93	12:00	36.64
5	07/07/93	22:00	42.59
6	10/07/93	14:00	45.11
7	23/07/93	13:00	48:12
8	13/08/93	13:00	49:10

Според извршените мерења на мерниот профил 002 се утврди дека е потребно појачување на подградбата. Со внесувањето на измерените вредности под реден број 4, системот алармира бидејќи се надминати максимално дозволените конвергенции и бара поставување на појака подградба.