2. gyakorlat: Követelmények ellenőrzése

A gyakorlaton a követelmények nyilvántartásával és ellenőrzésével kapcsolatos feladatok közül vizsgálunk meg néhányat.

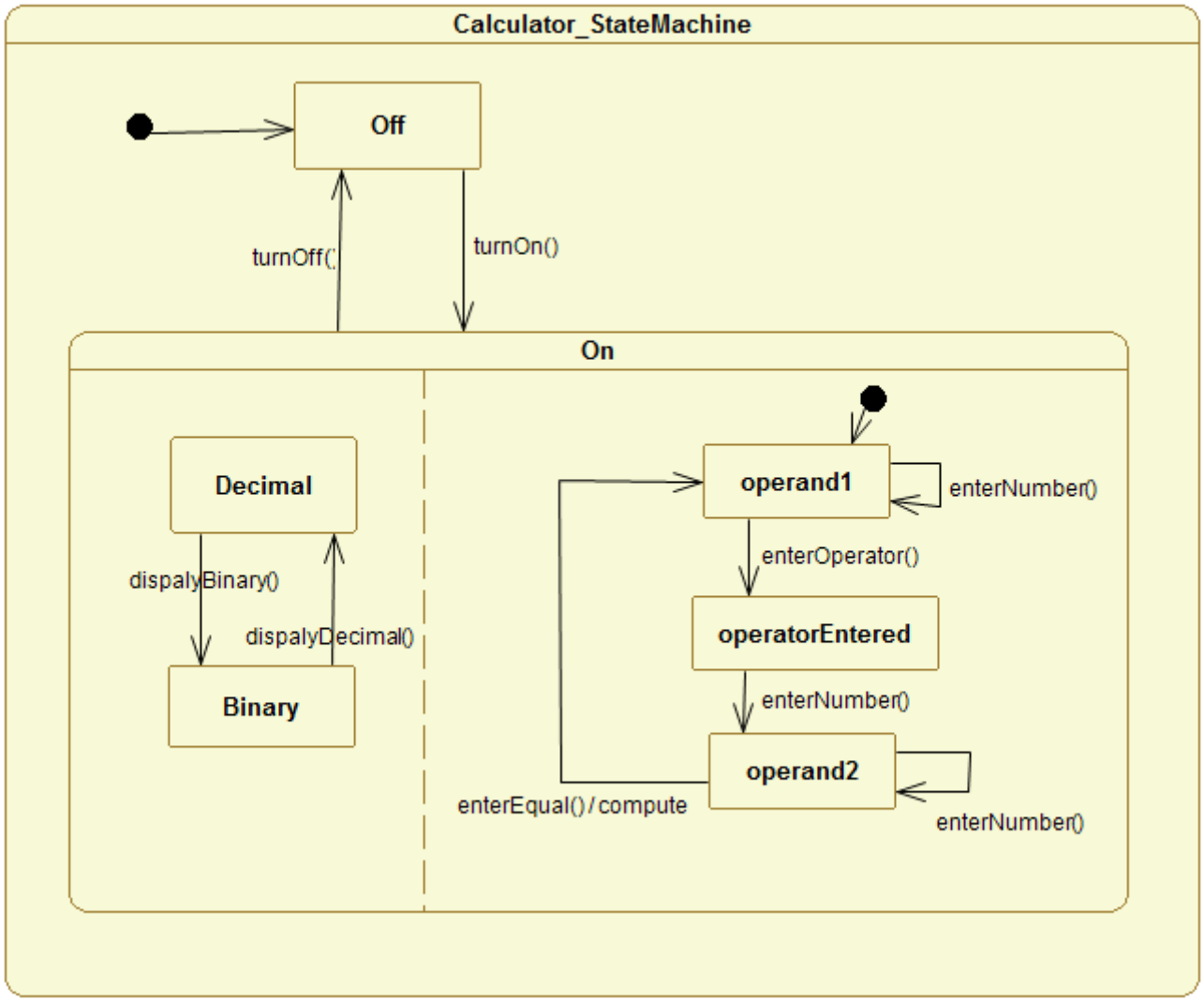
Követelményspecifikáció ellenőrzése átvizsgálással

A gyakorlati anyagok között megtalálható egy egyszerű számológép alkalmazás felületes követelményspecifikációja.

1. Átolvasással ellenőrizzük a leírást. Milyen hibákat találtunk, mit javasolnánk a hibák javítására?

Állapotgép modellek ellenőrzése

Az előző feladatban megvizsgált alkalmazás specifikálása során a következő lépésként elkészült a rendszer UML állapotgép modellje.

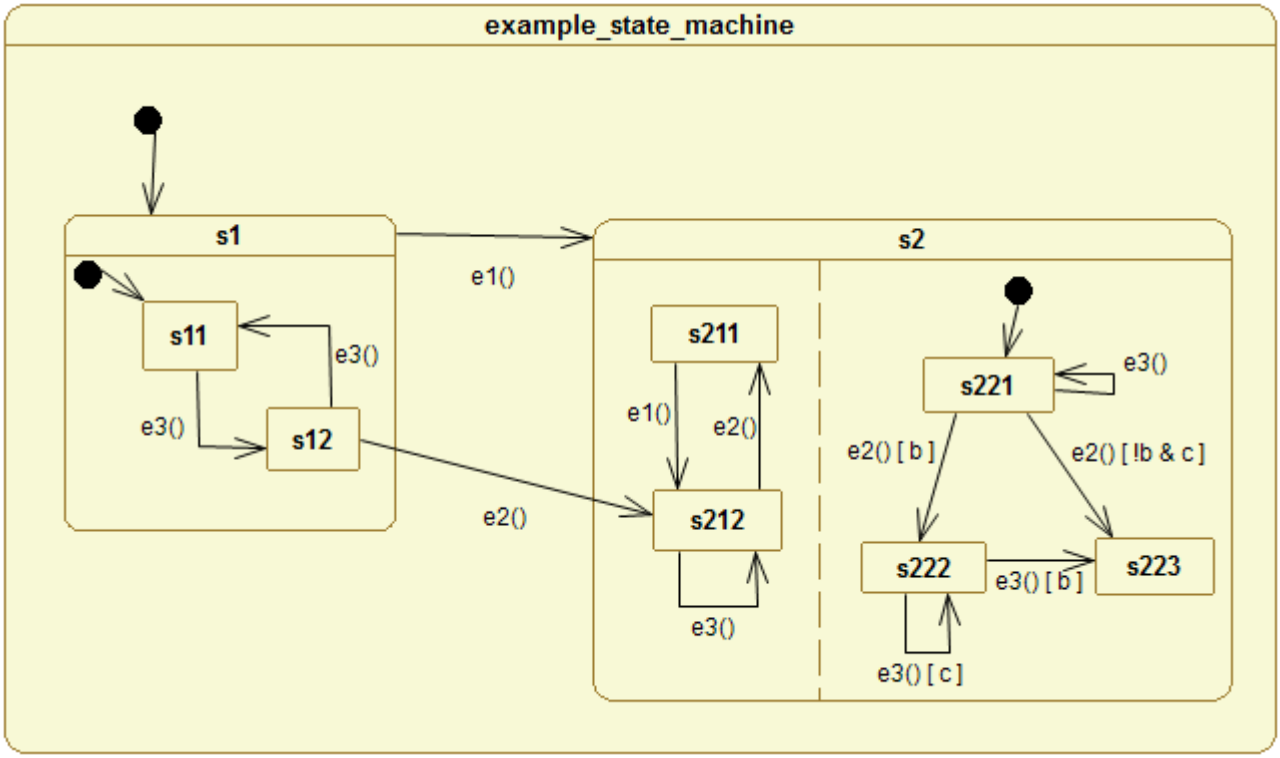


Az előadáson ismertetett állapotgép modell vizsgálati szempontok alapján ellenőrizzük az elkészült modellt.

2. Milyen hibákat találtunk?
    a. Ha teljessé szeretnénk tenni az állapotgépeket, a hiányzó tranzíciókat hogyan vennénk fel? Van-e olyan, ami nem egyértelmű?

3. Az UML állapottérképek ellenőrzése során bonyolultabb szituációkkal is találkoztunk. Vizsgáljuk meg az alábbi komplexebb modell teljességét és ellentmondás-mentességét!

Javaslat: a teljesség vizsgálatához készítsünk egy mátrixot a rendszer konfigurációival és eseményeivel.



Követelménykezelő eszközök: Rational DOORS

A gyakorlat második felében egy komplex követelménykezelő eszközt fogunk kipróbálni. A gyakorlaton rendelkezésre álló idő nyilvánvalóan nem elégséges az alkalmazás összes funkciójának kipróbálása, a cél csupán annyi, hogy egy-két alapfunkció megismerésével lássuk, hogy milyen többletet nyújt egy ilyen eszköz a kézi (Word, Excel dokumentumok) követelménykezeléshez képest.

Megjegyzés: A Rational DOORS-nak most csak az egygépes kliensét fogjuk használni, a több felhasználós szerkesztést, webes felületet, stb. biztosító szerver komponenst nem.

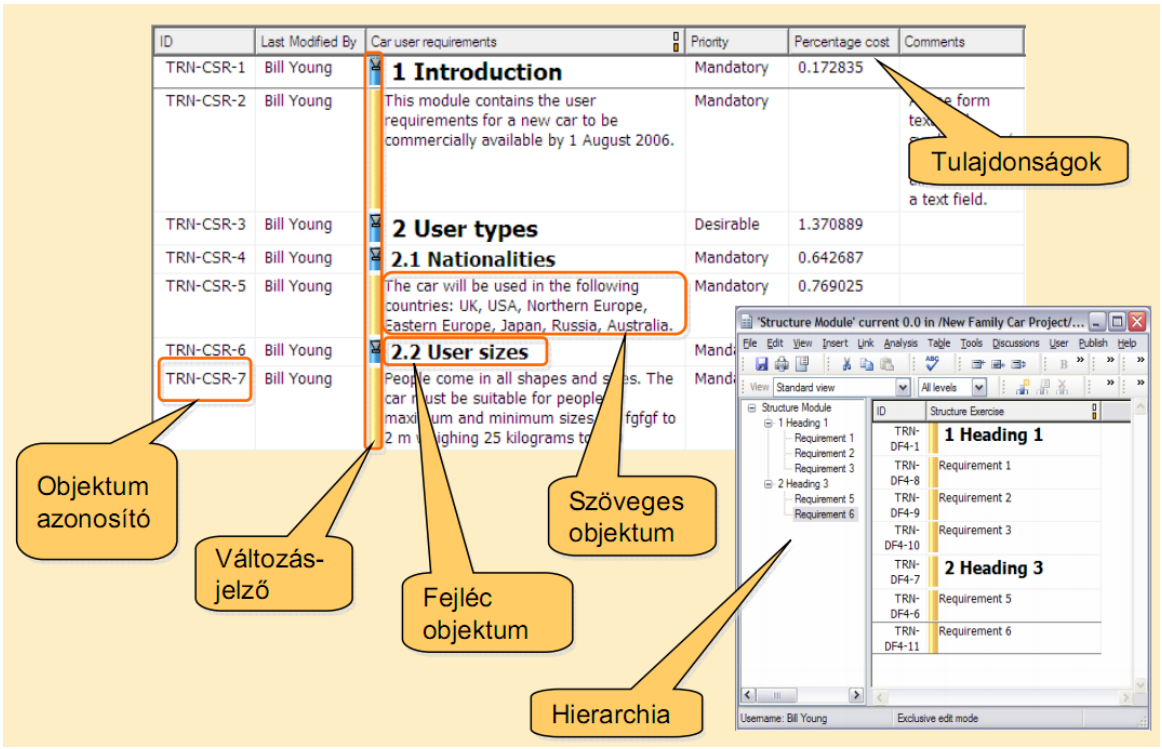
1. Indítsuk el a *C:\vmware-images\oktatas\SZET\IBM-Rational-DOORS_Client* virtuális gépet.
2. Ha megkérdezi, hogy mit csináljon a megváltozott géppel, akkor az **„I moved it”** opciót válasszuk.
3. Lépjünk be a virtuális gépre a mérésvezető által megadott jelszóval, majd indítsuk el az asztalon található *IBM Rational DOORS 9.2 Example Data* parancsikont.
4. Lépjünk be a DOORS-ba az *'Eric McCall'* felhasználóval (jelszó: *Training*)

A gyakorlaton először a DOORS beépített minta projektjét fogjuk használni. Ez egy új autó kifejlesztésével kapcsolatos követelményeket gyűjti össze.

5. Navigáljunk el a bal oldali fa struktúrában a következő projekthez: *Training / Example Data / Company Programs / Vehicle projects / Light Trucks / Prototypes / Copy of Sports utility vehicle*. Ezzel fogunk ezután dolgozni.

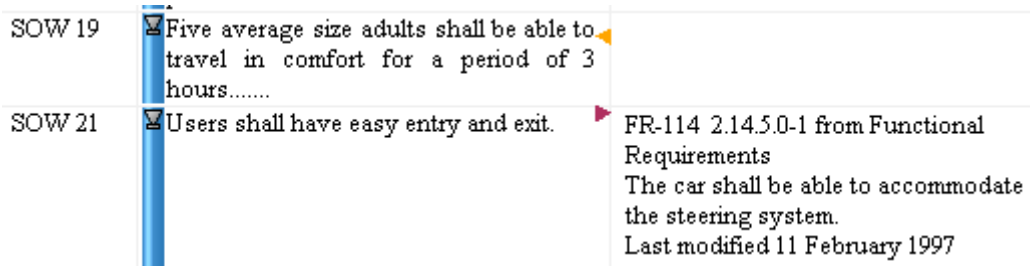
A DOORS alapvető elemei

A DOORS elemek alapvető szerkezete a következő. Egy projekten belül lehetnek könyvtárak, azon belül pedig modulok. Egy modul lehet *formal module*, mely egy általános objektum tár tulajdonképpen, vagy pedig egy *link module*, ami *formal module*-okban szereplő elemek kapcsolatát mutatja meg. Ha megnyitunk egy *formal* modult, akkor annak az általános szerkezete az alábbi ábrán látható.



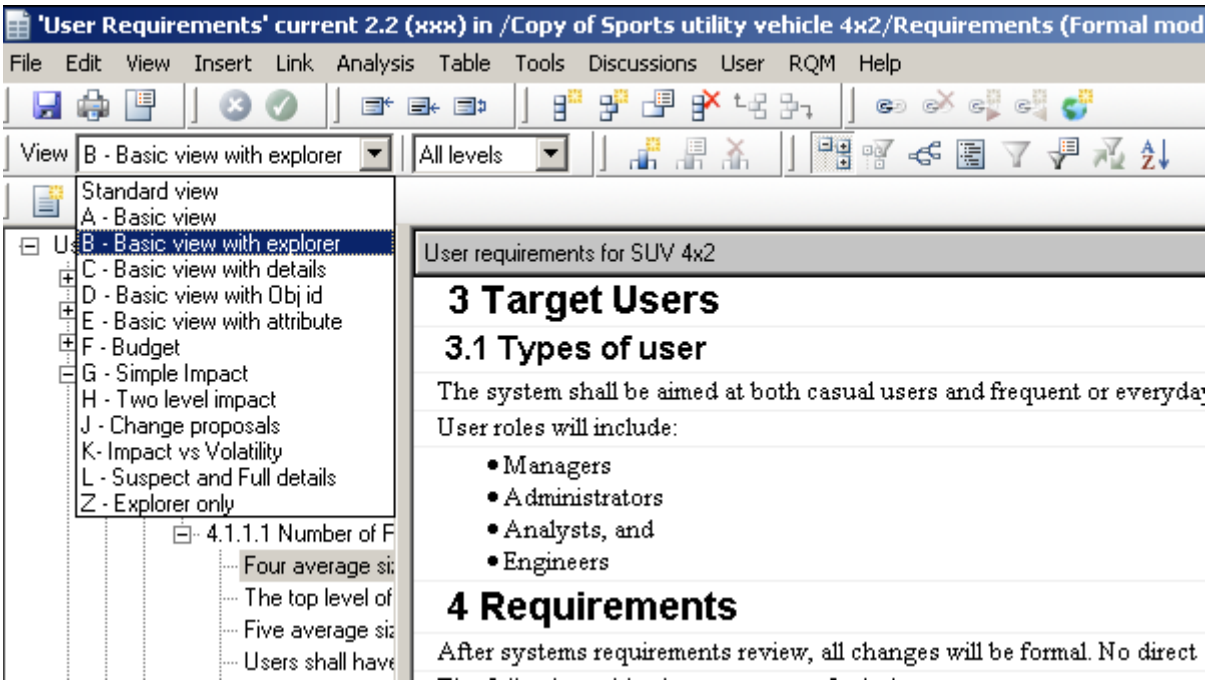
1. ábra: Egy DOORS modul elemei

Tehát az objektumoknak van *azonosítójuk*, és ezen kívül tetszőleges számú és típusú *attribútumuk*. Az objektumok hierarchiába szervezhetőek. Fontos még a változást jelző csík. Ez színkódolással jelzi az objektum állapotát (új objektum, módosítva volt, stb.). Ezen kívül az objektum jobb oldalán kis nyilak jelzik, ha létezik innen induló vagy ide mutató link. A nyílra jobb gombbal kattintva tekinthetjük meg a hivatkozott elemet.



2. ábra: Linkek megjelenítése

Hogy az attribútumok közül melyiket jeleníti meg a felület, azt a kiválasztott nézet dönti el.



3. ábra: Nézet kiválasztása

Az attribútumok lehetnek szerkeszthető típusok vagy pedig a DOORS saját DXL script nyelvén megfogalmazott lekérdezések. Ezek alapján látható, hogy egy kellően rugalmas, az adott projektnek megfelelő rendszert lehet kialakítani.

Egy egyszerű követelmény leírás

Nézzük meg akkor a példa adatbázisban szereplő leírásokat.

6. A *Requirements / Use Cases / Car Use Scenario* modul tárolja az autóval kapcsolatos felhasználói eseteket és alap forgatókönyveket.
    a. Figyeljük meg a követelmények szerkezetét! Pl. ezek mind kötelező követelmények, így az angolban a shall igét kell használni ennek a kifejezésére.
    (Lásd http://standards.ieee.org/guides/style/section5.html)

b. Vegyünk fel egy új elemet a 2.1-es pontba, mely szerint a felhasználónak meg kell tudnia állapítania a kerekek nyomását is.

   i. Egy új objektum (az eszköztár megfelelő gombját használva) alapértelmezés szerint megfelelő szintű címsorként jelenik meg. Amennyiben szöveges követelményt szeretnénk szerkeszteni, akkor az eszköztár megfelelő gombjával a címsor és a szöveges tartalom felcserélhető.

   ii. Másik megoldás lehet, hogy az objektum tulajdonságai között nem a *Heading*, hanem az *Object text* mezőt töltjük ki.

c. Nézzük meg az újonnan felvett objektum tulajdonságait. Milyen alapértelmezett attribútumok tartoznak még az objektumhoz?

7. A DOORS és a hozzá hasonló követelménykezelő rendszerek erőssége, hogy a követelmények változását és történetét lehet követni, minden módosítást vagy kapcsolódó megbeszélést lehet az egyes elemekhez kötni.

   a. Nézzük meg az *US10* azonosítója objektum történetét (a *Properties* ablakban). Kik módosították az objektumot?

   b. Az objektumhoz tartozó megbeszéléseket, megjegyzéseket a *Discussion* résznél lehet megnézni. A követelménnyel kapcsolatban pontosítást kértek, válaszoljunk a megjegyzésre.

8. Ezekből a használati esetekből és forgatókönyvekből el lehet kezdeni származtatni a követelményeket. A *Requirements* mappában szerepel egy *User Requirements* és egy *Functional Requirements* modul. Nézzük meg mindkét modul tartalmát. Milyen típusú követelmények kerültek az egyikbe és a másikba? Milyen típusú kapcsolat van a két modul között?

   a. A két modul elemei között már vannak linkek, melyek a követhetőség megvalósításában segítenek. Nyissuk meg a *Functional Requirements* modult, és válasszuk az *Incoming Links* nézetet (ha még nem az lenne kiválasztva).

   b. Keressük meg és nyissuk meg az FR-7 követelményhez tartozó felhasználói követelményt.

   c. Az SOW 29-es felhasználói követelmény („Users shall be able to travel automatically at predefined speeds.”) kapcsolódik az US29-es Use Case Scenariohoz. Jelezzük ezt egy link létrehozásával is (*Start Link*, majd a cél kiválasztása és *Make link from start*).

   d. A linkeket jobban át tudjuk tekinteni, ha a megfelelő nézetet választjuk. Nyissuk meg a *User Requirements* modult, és pl. használjuk a *Simple Impact* nézetet.

9. A projektnek nemcsak a követelményeit, hanem a többi elemét is be lehet importálni a DOORS rendszerbe.

   a. A *Project plans* könyvtárban a projekt MS Project tervéből importált elemek, a *Design* és *Implementation* könyvtárban az autó komponensei és a hozzájuk tartozó program részletek találhatóak. Így ezekhez is lehet linkeket definiálni, majd a hozzájuk tartozó követelményeket megkeresni.

   b. A *Test and Verification* könyvtárban néhány ellenőrzési cél és lépés van definiálva. A *Verification Methods* modult nézzük meg az *Evaluation* nézetében. Keressük meg, hogy a *Test 12* melyik felhasználói követelményhez tartozik!

10. Egy ilyen rendszer akkor hasznos, ha tudunk meglévő dokumentációkat importálni bele, és a követelmények elkészítése után a szükséges dokumentumokat exportálással elő tudjuk állítani.

    a. Exportáljuk ki a felhasználói követelményeket HTML dokumentumba.

11. A követelményekkel kapcsolatos változtatásokat érdemes egy jól definiált folyamat alapján kezelni. A DOORS-ban erre való a *Change Proposal System*. Itt módosítási javaslatokat küldhetünk be modulokhoz, majd azokat elfogadhatjuk, elutasíthatjuk, stb.
    a. Nyissuk meg a projekt gyökerében lévő *Change Proposal System* könyvtárból a *Suggestions* modult. Zárjuk le az S-10 azonosítóval rendelkező javaslatot (fogadjuk el vagy utasítsuk el, attól függően, hogy egyetértünk-e vele).

Saját DOORS követelmény projekt létrehozása

Most, hogy megismertük a DOORS alapvető funkcióit, készítsük el a CALCULATOR alkalmazás követelmény leírását DOORS-ban.

12. Hozzunk létre egy CALCULATOR nevű projektet.
13. Hozzunk létre egy általunk jónak látott könyvtár és modul struktúrát, majd töltsük fel a szöveges dokumentumban szereplő követelményekkel.