

KUTATÓI BESZÁMOLÓ

**Krónikus hemodialízisgép dializáló oldat  
előkészítést vezérlő szoftverének  
objektumorientált szemléletű fejlesztése**

Horváth Gábor

Gyakornok, B.Braun Medical Kft.

2015.01.28.

## Krónikus hemodialízis

A vese bab alakú páros szerv, melynek elsősorban a kiválasztásban van fontos szerepe, úgy mint a vízdékony anyagcsere-végtermékek eltávolításában, a só- és vízháztartás szabályozásában, valamint a fehérjék és nukleinsavak bomlástermékeinek kiválasztásában. Amennyiben a vesék nem működnek megfelelően, úgy nem képesek ezen feladatok maradéktalan ellátására. Súlyos esetben transzplantációra van szükség az életben maradáshoz, ám erre nincs mindig lehetőség. Azt az eljárást, melynek során a veseelégtelenséggel rendelkező betegek vére a különféle salakanyagoktól megtisztításra kerül, dialízisnek hívjuk.

Alkalmaznak testen belüli és testen kívüli dialízis módszereket is, ám utóbbi jóval elterjedtebb. Testen kívüli módszerek esetén számos típus ismeretes, melyek közül az egyik legsűrűbben alkalmazott módszer a hemodialízis. Megkülönböztetjük akut és krónikus fajtáját, melyekben közös, hogy orvosi felügyelet mellett végzik dialízisközpontokban. Akut veseelégtelenség esetén csak véges ideig szükséges a kezelés folytatása, mert a vesék működése idővel helyre áll, viszont krónikus esetben a páciensnek élete végéig szüksége van a terápia alkalmazására. Egy-egy ilyen kezelés akár 4- 5 órán át is tarthat a terápia paramétereitől függően és hetente átlagosan 3-4 alkalommal végzendő.

## A jelenlegi helyzet

A jelenlegi krónikus hemodialízisgépben alkalmazott szoftverarchitektúra nem alkalmas hosszú távon hatékony továbbfejlesztésre. A különböző funkcionalitások karöltve, egymásra támaszkodva valósítják meg feladatukat, valamint az egyes mechanizmusok lezajlása nehezen követhető végig a szoftver egészén. A kellő fokú modularizálás hiánya egyrészt megnehezíti a csoportos fejlesztést, másrészt az egyes szoftverkomponensek a későbbi termékek esetén nem, vagy csak jelentős módosítással lesznek újrafelhasználhatóak.

Az iparban több olyan beágyazott környezetben használatos szoftverarchitektúra is elterjedt, amelyek megoldást nyújtanak ezen problémákra. Számos tekintetben is hatékonyak tűnnek az úgynevezett referencia architektúrák, amelyek esetén az adott problémát szoftverkomponensekre bontjuk és definiáljuk azok kapcsolatait. A modularizálásnak köszönhetően az egyes feladatok jól szétválasztottak, amely számos előnyt jelent. A hordozhatóság és újrafelhasználhatóság mellett egy ügyes megtervezés esetén az egyes komponensek védettek lesznek egymással szemben. A rendszer egészének karbantarthatósága sokkal jobb, mint a jelenlegi architektúra esetén. A kialakítás lehetőséget biztosít közös elnevezés-használatra, amely egyértelmű, letisztult és kényelmes fejlesztést tesz lehetővé. A szoftvermodulokat logikai rétegekbe rendezve az architektúra határvonalai jól láthatóak, a köztes interfészek jól definiálhatóak. A referencia architektúra azonban néhány hátrányt is magával hordoz, amelyeket nem lehet figyelmen kívül hagyni. Beágyazott környezetben, azon belül is orvosi készülékekben elengedhetetlenül fontos szempont a real-time működés, amelyet az architektúra önmagában nem feltétlenül garantál, komoly odafigyelésre van szükség annak tervezése során. Amennyire sok előnnyel rendelkezik, olyan gondos megtervezést igényel. Ehhez olyan emberek szükségesek, akik komoly szaktudással

rendelkeznek és ismerik a pontos feladatot, amelyet aztán képesek szétszedni megfelelő komponensekre. Az architekt feladata igen komoly, hiszen ha túl későn derül ki, hogy a funkciók szétválasztását másképp kellett volna megoldani, akkor annak orvoslása igen nagy energia- és anyagi többlettel jár.

Az iparban a referencia architektúrák csoportjában két jeles képviselő is szerepel, nevezetesen az Android és az AUTOSAR. Ha teljes egészében nem is, de ésszerűen alkalmazva a belőlük levont tanulságokat és előnyöket, egy saját architektúra kialakítására van lehetőség, ami igen jól illeszkedik a dialízisgépek működéséhez.

## Az új módszer

Felmerült az ötlet, hogy tanulva a korábbi esetekből, egy jól átgondolt megoldásra kéne áttérni a dialízis gépek szoftverfejlesztését tekintve. Vizsgáljuk meg, hogy miket várunk az új módszertől:

- Fejlesztési idő csökkenése
- Átláthatóbb szoftveres megoldások
- Újrafelhasználható szoftverkomponensek
- Könnyebb karbantarthatóság, tesztelhetőség
- Orvosi követelményekhez könnyen igazodó szoftverfejlesztés
- Automatizált megoldások a fejlesztés során

A PC-s világban manapság nagyon elterjedt az úgynevezett objektumorientált programozás (OOP), amely számos előnnyel rendelkezik a korábban alkalmazott megoldásokhoz képest. A probléma az, hogy beágyazott rendszerek esetén nem egyértelmű az alkalmazása, ám mindenképpen valami ehhez hasonló koncepciót lenne célszerű követni.

A fentebbi szempontokat figyelembe véve egy olyan megoldás született, amellyel a kitűzött célok megvalósíthatóak. Az OOP kapcsán általában a jelenleg legelterjedtebb programozási nyelvek és platformok jutnak eszünkbe, mint például a .NET vagy a Java. Ezen megoldások használata elvetésre került több okból is. Az egyik, hogy ezen platformok olyan kész megoldásokkal (előre megírt osztályokkal) rendelkeznek, amelyeket külön-külön tesztelni kéne, hogy orvosi eszközökben is használhatóak-e. A platform minden egyes újabb kiadásakor ellenőrzést kéne végrehajtani a módosításokra.

Az új szemlélet felállításakor a könnyű és gyors átláthatóság is fontos szempont volt, szóba jött tehát a model-based design gondolata is. Mindezeket figyelembe véve kézenfekvőnek tűnt tehát Matlab/Simulink-kel próbálkozni. Ez olyan szempontból hátrányt jelent, hogy meg kell tanulni a használatát, hiszen a szöveges programozási nyelvekhez képest más gondolkodást igényel. Simulinkkel való fejlesztés esetén nem használhatóak az eddig jól bevált programozási paradigmák és tesztelési módszerek, igaz az OOP szemléletet figyelembe lehet venni. Ennek ellenére érdemes megvizsgálni, hogy mit is biztosít számunkra a Simulink. Egyrészt „vizuálisan” kódolhatunk, nem veszünk el a hosszú, bonyolult kódsorokban. Színekkel, kiemelésekkel, valamint ügyes topológia alkalmazásával a fejlesztő által létrehozott szoftvertermék működése hamarabb átlátható, ezáltal az egyes fejlesztői

hibák felderítése is kevesebb időt vesz igénybe. Az elkészített Simulink modellből lehetőségünk van pár kattintással automatikusan C kódot generálni, amely ez után futtatható a dialízisgépen. Természetesen egy sajátos gondolkodásmódot igényel Simulink-ben fejleszteni, mint más programozási nyelveken, ez azonban nem jelent semmiféle hátrányt. Az OOP szemléletben megjelenő újrafelhasználhatóságot, azaz a kódDuplikáció elkerülését Simulink library-k segítségével valósíthatjuk meg. Minden olyan funkcionalitást, amelyet több, mint egy helyen szeretnénk használni, library-ként kell létrehozni. Ezt követően minden olyan helyen, ahol szükség van ezen funkcionalításra, egyszerűen csak belinkeljük a korábban létrehozott library-t. Egy esetleges módosítás mindenhol megjelenik a szoftverben, ahol alkalmaztuk a linkelést. A Matlab további nagy előnye, hogy nagyon kényelmesen lehet excel és csv fájlokkal együttműködni, valamint, hogy számos automatizált megoldást lehet alkalmazni a platform által szolgáltatott scriptnyelv segítségével, amelynek használata könnyen tanulható.