

# A földi ellenőrző berendezésekben alkalmazott programozási technikák

BALAJTHY KÁLMÁN

KFKI Részecske és Magfizikai Kutatóintézet, balajthy@rmki.kfki.hu

LIPUSZ CSABA, SÓDOR BÁLINT, SZALAI SÁNDOR

SGF Kft., {lipusz.csaba, soba}@freemail.hu, szalai@sgf.hu

**Kulcsszavak:** úrkutatás, földi ellenőrzőberendezés, elosztott adatgyűjtő- és vezérlő-rendszerek, beágyazott rendszerek

*Az elektronikus földi ellenőrzőberendezés (EGSE) alapvető feladata az úrkutatási műszerek tesztelése és az űrszonda elektromos illesztő felületeinek szimulálása a fejlesztés különböző fázisaiban. A magyar fejlesztők húsz éves tapasztalattal rendelkeznek az EGSE létrehozásában és alkalmazásában. A cikk az elmúlt években az EGSE rendszerek fejlesztése kapcsán felhalmozott tapasztalatokat, az alkalmazott technológiákat mutatja be röviden.*

## 1. Bevezetés

A tudományos úrkutatási programok jellemzően nemzetközi együttműködésben készülnek, a misszió során használatos különböző tudományos műszerek, illetve vezérlő és adatgyűjtő egységek fejlesztése párhuzamosan folyik, különböző országok kutatóhelyein. Természetesen a fejlesztés során a különböző fejlesztő részlegeknek nem állnak rendelkezésükre a máshol készülő, szintén még fejlesztési stádiumban lévő komponensek, noha a fejlesztés során erre, kiváltképp a vezérlő és adatgyűjtő egységre nagy szükség van. Ennek a problémának a feloldása az úgynevezett elektronikus földi ellenőrzőberendezések (Electrical Ground Support Equipment – EGSE) segítségével történik. Az EGSE rendszerek biztosítják a műszer az űrszondára szerelt tesztkörnyezetét, a különböző komponensek interfészének szimulációját a fejlesztés és minősítés során. A Részecske és Magfizikai Kutatóintézet (KFKI RMKI) már hosszú évek óta részt vesz úrkutatással kapcsolatos fejlesztésekben és az intézethez kapcsolódóan az SGF Kft. is. A KFKI RMKI elsősorban fedélzeti részek, míg az SGF Kft. elsődlegesen a földi ellenőrzőberendezések fejlesztésében vállal szerepet. Az intézet és a kft az elmúlt húsz év során több jelentős feladatban való sikeres szereplése révén jó tapasztalatra és jó referenciára tettek szert.

Az EGSE rendszerek rövid történelme és néhány megvalósított, illetve megvalósítás alatt álló rendszer bemutatása után sorra vesszük az alkalmazott rendszerprogramozás technikai megoldásait, amelyek megkönnyítik a fejlesztési időszakban még gyakran változó paraméterek megjelenítésével kapcsolatos problémák minimalizálását és lehetővé teszik az automatikus tesztlekés futtatását.

## 2. Elektronikus földi ellenőrzőberendezések

Az első EGSE rendszerek még dedikált hardverből és szoftverből épültek fel, majd az asztali számítógépek számítási kapacitásának növekedésével lehetővé vált

a PC-k alkalmazása mind a szimulációs, mind az adatgyűjtő alrendszer funkcióinak megvalósítására. A nyolcvanas években a PC-k ISA buszára kifejlesztett egyedi illesztő kártyák szimulálták az űrszondák különböző jeleit. Ekkor a PC erőforrásai maximálisan kihasználásra kerültek, például a PC memóriája közvetlenül fogadta a műszer mérési adatát a közvetlen memória hozzáférés útján. A PC ezen memóriatartománya volt egyben a megjelenítés forrása is, a szoftverek a DOS rendszer alatt készültek, így a grafikus megjelenítés biztosítása jelentős szoftvermunkát jelentett.

Ebben az időben készült a KFKI RMKI-ban a Szovjet Űrügynökség Phobos szondájának több kísérletéhez, majd a jelenleg is a Szaturnusz környezetében működő NASA Cassini szondájának két műszeréhez az EGSE. Napjainkban az inetrált áramkörök viharos fejlődése, valamint az operációs rendszerek javuló szolgáltatásai lehetővé teszik a hatékonyabb EGSE-k létrehozását, így lehetővé válik funkcióinak fizikai szétválasztása is. A pontosabb eredmények érdekében az űreszközök szimulációját valós idejű beágyazott processzoros egységekkel valósítjuk meg, míg adatgyűjtésre, adatfeldolgozásra, továbbá a felhasználói felület megvalósítására (Graphical User Interface – GUI) a széles körben elterjedt sokfunkciós kereskedelmi számítógépeket használjuk.

### 2.1. ASPERA

Az Európai Űrügynökség Vénusz-kutató missziójának kapcsán, a VenusExpress űrszonda ASPERA-4 tudományos műszerének kalibráló rendszerének automatizálását vállalta a magyar fél. A feladat kapcsán egy olyan rendszert kellett megvalósítani, amely a műszer kalibrációs folyamatát felügyeli. A feladatot megoldása során egy elosztott intelligenciájú, három önálló beágyazott processzoregységet és egy kereskedelmi forgalomban beszerezhető számítógépet magába foglaló rendszer jött létre [3]. Az utóbbin Windows, míg a beágyazott processzorokon valósídejű Linux operációs rendszer fut. A beágyazott rendszerek feladata a kalibrációs környezet vezérlése és a mérőműszer vezérlése, valamint a

mért tudományos adatok összegyűjtése. A laptop vagy asztali számítógép feladata a teljes rendszer felügyelete, felhasználói interfész biztosítása a kalibrációs folyamat felügyeletéhez és az összegyűjtött adatok vizuális megjelenítése az eredmények kiértékeléséhez.

## 2.2. Plazma Hullám Mérőrendszer

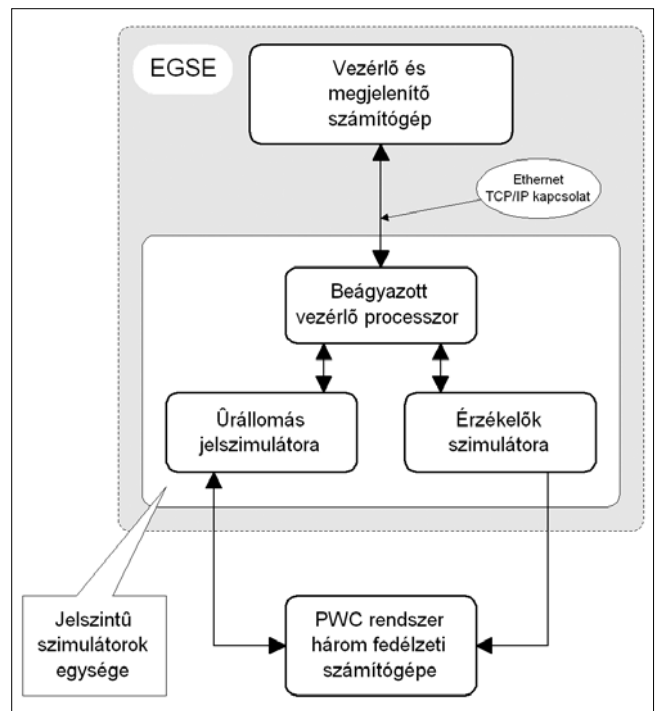
A nemzetközi űrállomás orosz szegmensébe kerülő Plasma Wave Complex (PWC) kísérlet kapcsán a magyarok feladata az érzékelők adatgyűjtő rendszerének és annak földi ellenőrzőberendezésének megtervezése és implementálása volt. A PWC adatgyűjtő és vezérlő rendszere három számítógépből áll [4] (1. ábra).

A kísérletben 11 tudományos műszer vesz részt, ezek az űrállomás különböző pontjain végeznek méréseket és az eredményeket két az űrállomás külsőoldalon lévő beágyazott processzor ethernet TCP/IP kapcsolatán át továbbítja a beltéri központi adatgyűjtő berendezésnek. A központi adatgyűjtő feladata az, hogy a mérési eredményeket archiválja és a földről érkező parancsokat a műszereknek továbbítsa. Az EGSE feladata a fejlesztés korai szakaszában a még el nem készült tudományos berendezések interfészeinek szimulációja, későbbi szakaszokban, pedig a rendszer egység szintű és integrációs tesztelésének támogatása. A fejlesztendő rendszerek között itt is megtalálható egy kereskedelmi számítógép, amelynek a feladata, hogy grafikus felhasználói felületet biztosítson a rendszer vezérléséhez, illetve a tudományos adatok megjelenítéséhez.

## 2.3. BepiColombo

A BepiColombo az európai űrügynökség Merkurt kutató missziója. A projekt kapcsán az intézet feladata az űrszonda Planetary Ion CAMera (PICAM) műszeréhez EGSE rendszer implementálása. Az EGSE feladata a kamera üzemmódjainak és pozíciójának vezérlése, valamint a műszer által készített képek fogadása és archiválása. Az EGSE rendszerekben folyó kommunikáció aszimmetrikusságára jellemző módon az EGSE-től a kamera irányában folytatott kommunikáció (parancsok vagy telecommand) sávszélessége várhatóan kisebb lesz mint 4 kbit/sec, míg ellenkező irányban (mérési eredmények vagy telemetria) 30 és 675 kbit/sec között változhat az adatforgalom, a Földtől való távolság függvényében.

A fejlesztés alatt álló rendszer, a már elkészített architektúraterv szerint, egy beágyazott és egy kereskedelmi számítógépből fog állni. A korábbi rendszerekhez hasonlóan itt is a beágyazott rendszer feladata, hogy a műszer felé az űreszköz interfészeinek a szimulációját végezze, a számítógép feladata az adatmegjelenítés és a felhasználói felület biztosítása a vezérléshez. A számítógép és a beágyazott rendszer közti kommunikáció ethernetet keresztül zajlik, a tudományos műszer és a beágyazott rendszer között pedig SpaceWire szabványú interfészen folyik a kommunikáció. A SpaceWire szabványt implementáló interfész kártya és annak real-time linux drivere a csoport munkatársainak fejlesztése. Ez gyakorlatilag készen van, a további feladat az alkalmazói szoftverek tervezésének, fejlesztésének folytatása.



1. ábra  
A PWC EGSE rendszerének vázlatja

## 3. EGSE rendszerek fejlesztésének programozás-technikai aspektusai

Az EGSE rendszerek implementálása kapcsán felmerülnek programozástechnikai kérdések is, mint például a rendszer párhuzamossága, megbízhatóság, valószínű működés. A továbbiakban ezen aspektusokból kerül bemutatásra az EGSE rendszereknél alkalmazott fejlesztési metodika.

EGSE rendszerekben a kiszolgálás minősége szempontjából többféle adatfolyamot lehet megkülönböztetni. Első és legfontosabb a nagytömegű, valós időben érkező, tudományos mérési eredményeket hordozó adatfolyam. A tudományos adatfolyam kezelés közben adat nem vesztethet el, hiszen ez később lehetetlenné tenné az adatok kiértékelését, feldolgozását. Ezeknek a kezelése magában foglalja a kiszolgálásukat és archiválásukat. A földi parancsokat tartalmazó adatfolyam prioritása magas, de adattartalma kevés, kismértékű várokozottatás, késés nem okoz problémát.

Az adatok feldolgozása, megjelenítése az egyik legalacsonyabb prioritású feladat egy EGSE rendszerben. Ennek oka, hogy egyrészt számítási igény szempontjából ez a legmegterhelőbb feladat, lévén a nagytömegű beérkező adatfolyamon kell végrehajtani és a grafikus megjelenítés, a különféle számítási feladatok időigényeinek. Másrészt a veszteségmentes archiválásnak köszönhetően, később 'off-line' módon elvégezhetőek ezek a műveletek a tudományos adatokon. Az utolsó funkció, ami adatfolyam kezelést igényel a rendszer és a tudományos műszer belső állapotainak illetve az eseményeknek a naplózása, ez nem tartalmaz számottevő mennyiségű adatot.

### 3.1. Elosztott feldolgozás

Az EGSE rendszerek feladata kettős, egyfelől valós időben kell adatgyűjtést és beavatkozást végezni egy olyan környezetben, mely szigorú ütemezési követelményeket támaszt a rendszerrel szemben. Másfelől a vezérlésbe való beavatkozáshoz, illetve az eredmények megjelenítéséhez egy grafikus felületet kell biztosítani a felhasználó felé. A grafikus megjelenítés, adatok archiválása, felhasználói interakciók fogadása nehezen összeegyeztethető egy valós idejű, szigorú ütemezéssel, így az implementált EGSE rendszerek minden esetben legalább két komponensből állnak. A befolyásolt, megfigyelt környezettel közvetlenül kapcsolatban álló rész valós idejű operációs rendszert futtató beágyazott processzorkártyás komponensekből áll, míg a grafikus megjelenítést egy asztali számítógép végzi. A párhuzamos feldolgozást végző egységek egymással lazán csatoltak, köztük régebben sorosporti, újabban a növekvő sávszélesség igény és rugalmasság miatt TCP/IP protokoll fölötti kommunikáció zajlik.

A feladat komplexitásától függően egy vagy több beágyazott komponens kerül alkalmazásra. A beágyazott egységek real-time linux operációs rendszeren futó, a konkrét feladathoz dedikáltan fejlesztett szoftvert futtatnak. Az alkalmazott operációs rendszer, már meglévő valós idejű-linux kernel modulokból, az intézetben került kialakításra, maximálisan szem előtt tartva a vele szemben támasztott követelményeket és a futási környezetet.

### 3.2. Adatátvitel

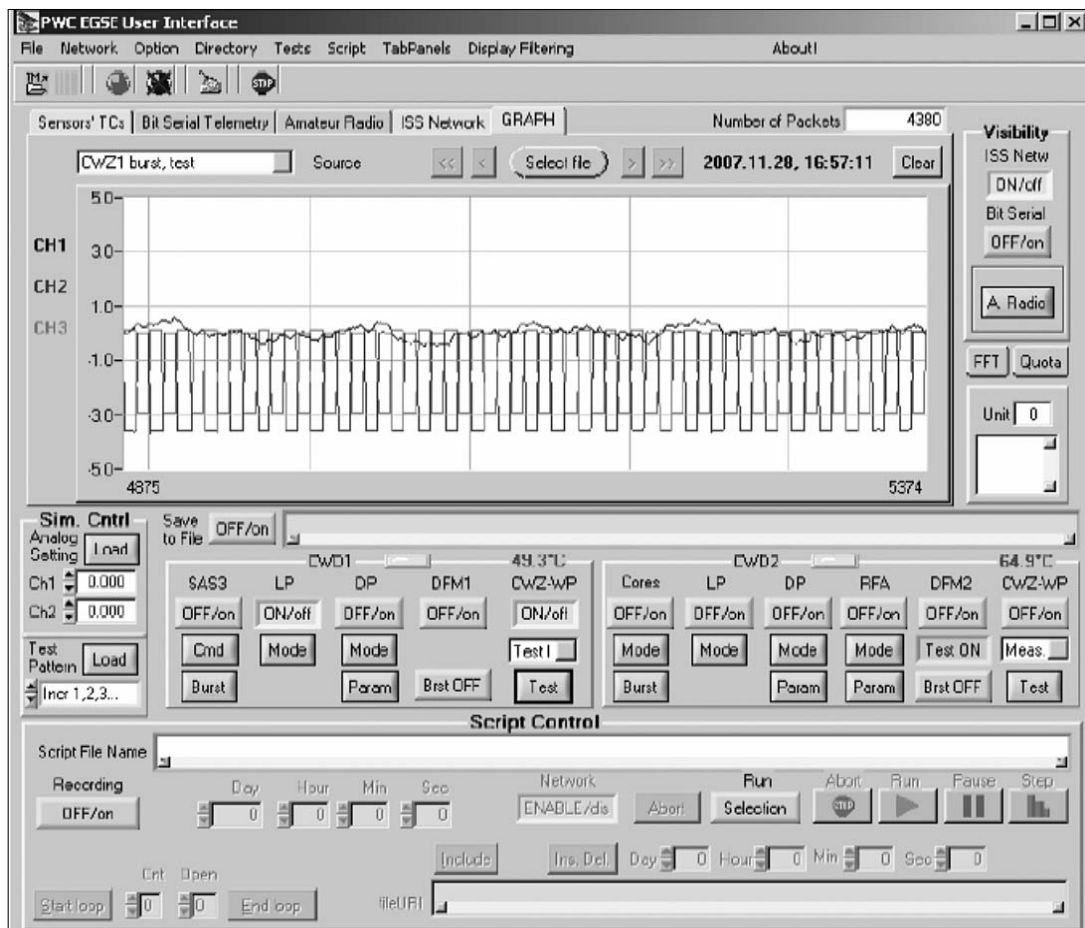
Az jelenlegi EGSE rendszer komponensei között az adatátvitel minden esetben egy adott fizikai rétegen, TCP/IP fölött implementált, dedikált kommunikációs protokoll szerint történik. A fizikai réteg lehet „hagyományos” utp-kábeles, fényvezetőkábeles vagy vezeték nélküli átvitel is. Az EGSE és a befolyásolt környezeti közötti kommunikációt mindig az utóbbi interfésze határozza meg, ennek teljes mértékben meg kell egyeznie az üreszköz megfelelő kommunikációs protokolljával. Ezen kommunikációs protokollok között egyaránt megtalálhatóak a hagyományos RS-232 sorosporti kommunikációs protokollok, a SpaceWire implementációk és újabban már az autópárból átvett CAN busz feletti kommunikációs protokollok is.

Az adatátvitelre jellemző az aszimmetrikusság, azaz jellemzően a 'downlink' (üreszköz felőli) irányban nagyságrendekkel nagyobb a forgalom, mint 'uplink' irányban (a felhasználói felülettől az üreszköz felé). Az üreszköz felől érkező adatok lehetnek tudományos mérés eredménye (tudományos adatok) valamint a műszer belső állapotait leíró adatok (HK vagy house-keeping adatok). Ezeket az adatokat veszteségmentesen kell eljuttatni a felhasználónak megjelenítés és archiválás céljából.

### 3.3. Grafikus kezelői felület (GUI)

A grafikus kezelői felület egy program, amely jellemzően Windows operációs rendszer alatt fut az asztali vagy hordozható számítógépen. Feladata az, hogy a fel-

2. ábra  
LabWindows/CVI-jal  
összeállított  
kezelőfelület a PWC  
EGSE projekthez



használót folyamatosan tájékoztassa a felügyelt rendszer és a teljes EGSE rendszer aktuális belső állapotáról, valamint beavatkozást biztosítson a vizsgálat alatt álló műszer működésében. A GUI feladata, hogy a tudományos műszertől érkező adatokat megjelenítse, illetve archiválja. Az aktuálisan futó projektekben a kezelőfelület feladatköre egy újabb jelentős funkcióval, a szcenáriókészítéssel bővült, ami lehetővé teszi, hogy egy adott forgatókönyvnek megfelelő felhasználói interakció sorozat rögzítésre kerüljön és később visszajátszható legyen.

3.3.1. Fejlesztőkörnyezet

A beágyazott rendszeren futó alkalmazások fejlesztése általában linux környezetben, karakteres szövegszerkesztő és gcc használatával történik. A GUI program fejlesztésére a National Instruments cég LabWindows/CVI terméke került kiválasztásra, a sokrétű grafikai szolgáltatása miatt. Az elmúlt évek során az EGSE rendszerek kezelői felületének fejlesztése során minden elvárásnak megfelelt. Grafikus komponenseinek köszönhetően hatékonyan támogatja úgy a tudományos adatok megjelenítését, mint a különböző portokon keresztül (hálózat, sorosport stb.) történő kommunikációt, valamint a többszálú programozást. A fejlesztő eszköz egy ANSI C alapú programozási környezetet és egy egyszerűen kezelhető, ablakok összeállítására alkalmas grafikus szerkesztőt biztosít.

3.3.2. Adatgyűjtés, megjelenítés

Az intézetben implementált EGSE rendszerekben az adatgyűjtés és megjelenítés minden esetben az általános célú számítógép (jellemzően Windows-t futtató gép) feladata. A nagymennyiségű beérkező adat feldolgozása időigényes feladat, míg a fogadásnak mindenképpen veszteségmentesnek kell lennie. A felada-

tok összeegyeztetéséhez többszálú program kerül implementálásra, ahol egy dedikált, minden feldolgozási folyamattól mentes külön szál (thread) végzi a TCP/IP kapcsolat kezelését, az adatok fogadását és a feldolgozó szál(ak)nak való továbbítást. A megjelenítést külön programszálak végzik, a szálak közötti kommunikáció a fejlesztőkörnyezet által felkínált, kifejezetten a párhuzamos elérés problémáit kiküszöbölő, úgynevezett szálbiztos üzenetsorokon (thread safe queue) keresztül történik. A 3. ábra a szálak közötti adatátvitelt vázolja. A beérkező adatot a TCP/IP kezelő szál fogadja és behelyezi a megfelelő pufferbe a csomag típusától függően. A pufferbe kerülő adat egy eseményt generál, ennek hatására a pufferhez regisztrál úgynevezett 'callback' függvény fog lefutni, a neki megfelelő feldolgozó szálaban.

A 4. ábra a megjelenítést végző számítógép és a beágyazott számítógép közötti TCP/IP kapcsolat létrehozását szemlélteti. A kapcsolat létrehozásakor egy új programszál jön létre, ebben fog futni minden a kommunikáció fenntartásáért felelős metódus. A szál feladata az, hogy fogadja az adatokat, valamint hogy a megfelelő adatfeldolgozó szálnak továbbítsa. További fontos feladata, hogy fogadás után ellenőrizze az adatcsomag konzisztenciáját, ami többnyire egy jól ismert bájt szekvencia (szinkronizációs minta) keresését és a csomaghossz ellenőrzését foglalja magában. Minden érvényes csomag, a fejlécének egy adott értéke alapján a megfelelő pufferbe kerül további feldolgozásra.

A szolgálati adatok (hőmérséklet, áramfelvétel, feszültség, állapotok) feldolgozása, azaz olvasható formában, a fizikai egységeinek megfelelő mértékegységben való átalakítása egy adott szolgálati adatcsomaghoz tartozó külső leírófájl alapján történik. Ez a programrész tulajdonképpen egy több leírófájl alapú interpreter.

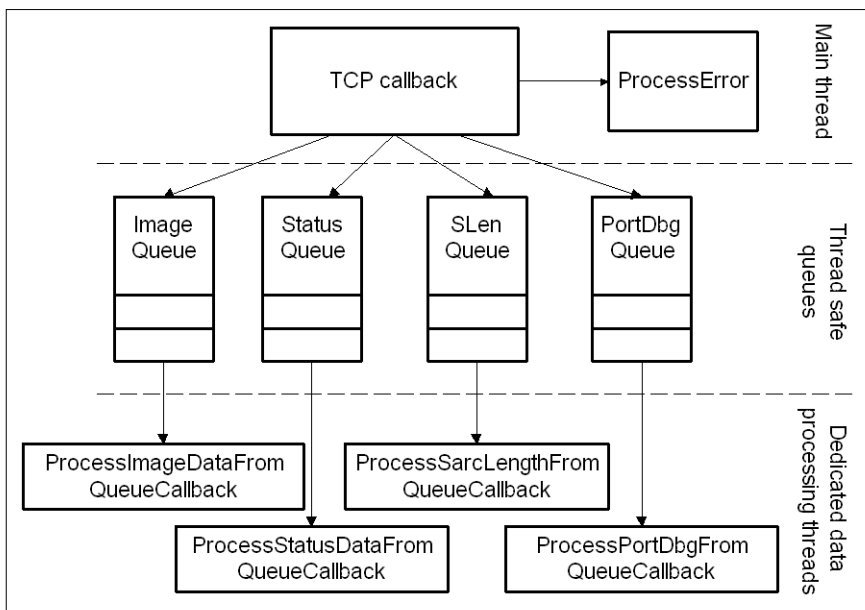
A fejlesztett EGSE rendszerek általában lehetővé teszik, hogy hozzájuk további adatkiértékelő modulok csatlakozhassanak. Ennek eredménye, hogy a tudományos műszer fejlesztői által a műszerből érkező ad-

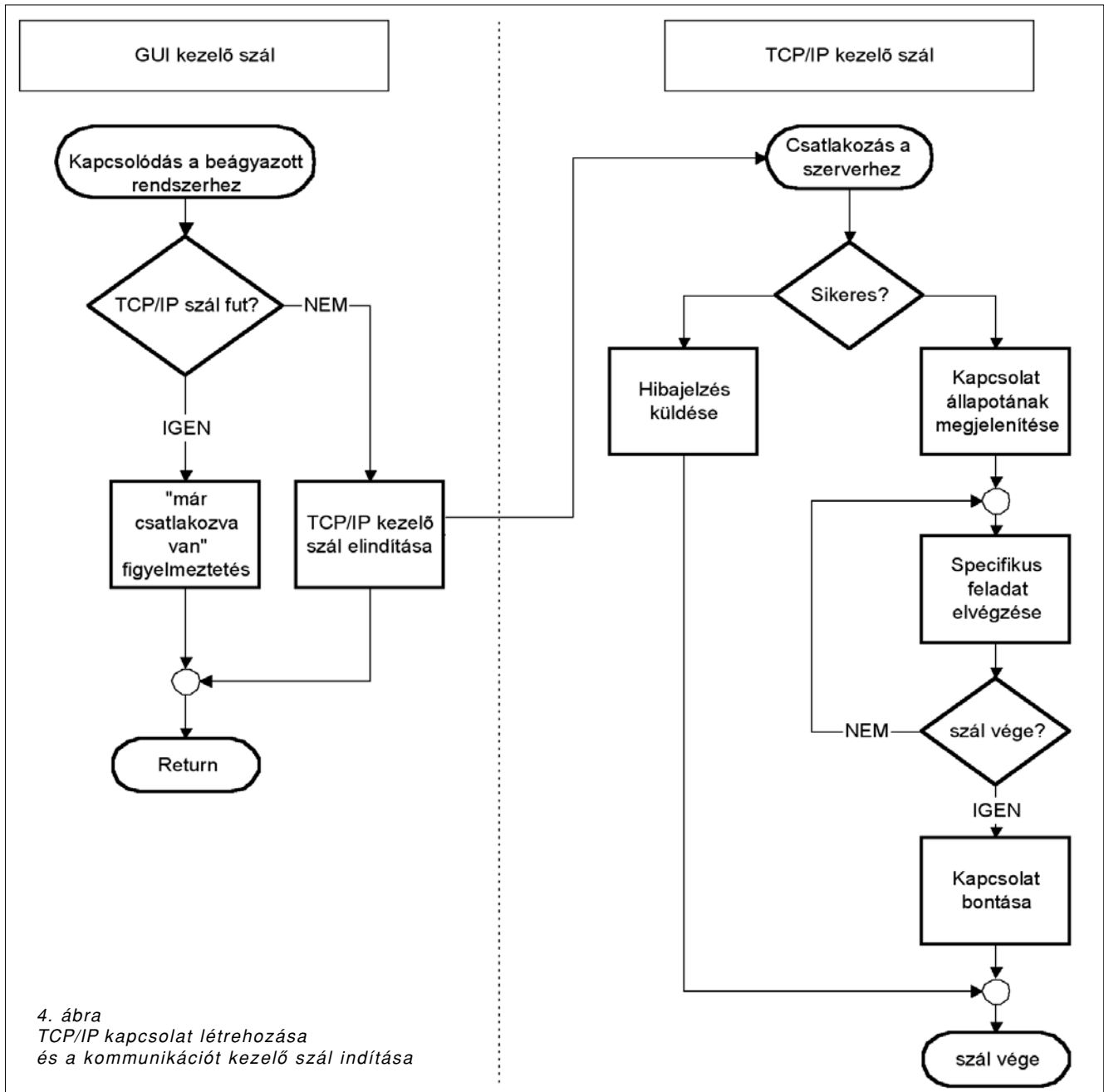
atok feldolgozásához implementált modul képes az EGSE rendszeren keresztül csatlakozni a műszerhez, parancsokat küldeni és fogadni. Az EGSE rendszer ekkor transzparens a külső feldolgozómodul számára.

3.3.3. Felhasználói interakció szekvenciák rögzítése

A felhasználó által a grafikus felületen végrehajtott interakció-sorozat rögzítésének két fontos célja van. Egyfelől egy adott forgatókönyvhöz illeszkedő, elmentett szekvenciák később tetszőleges időben-, akárhányszor visszajátszható, a mérés, tesztelés, kalibráció megismételhető lesz. Másrészt az elmentett szekvencia dokumentálja a szcenárió végrehajtását. Az igények, célok figyelembe-

3. ábra Szálak közötti kommunikáció 'thread safe queue-k' használatával





vételével a szekvenciák egy XML alapú szöveges fájlban kerülnek tárolásra [1,2]. Az XML szkript fájlok nyelvtana a mindig egy adott feladathoz, dedikáltan készül el. Az 5. ábrán egy ilyen szekvencia leíró (szkript) fájl részlete látható. Az XML fájlok írása és olvasása egy Xerces-C++ alapú DOM olvasót implementáló modul segítségével történik, a nyelvtan megadásához, bonyolultságtól függetlenül, XSD vagy DTD került felhasználásra.

### 3.4. Adatleíró réteg

Az utóbbi időben egyre nagyobb hangsúlyt kapott a fejlesztésekben, hogy az imple-

mentál rendszerek működésük során felhasznált, illetve előállított adatok olvashatóan, platformfüggetlenül kerüljenek leírásra. Ezen adatok lehetnek például a korábban tárgyalt forgatókönyveket leíró szkriptfájlok,

```

<Digout value="8"/>
<save act="on"/>
<acquisition act="on"/>
<acquisition act="off"/>
<acquisition act="on"/>
<delay value="1" dim="sec"/>
<acquisition act="off"/>
<save act="off"/>
<step axis="azimut" speed="5" value="1.000000" dir="pos"/>
<step axis="vertical" speed="5" value="1.000000" dir="pos"/>
<step axis="horizontal" speed="5" value="1.000000" dir="neg"/>
<turnPWR30supply act="off"/>
    
```

5. ábra  
Részlet egy XML szkript fájlból

vagy szimuláció esetén a szimulálandó egységek modelljeit leíró fájlok, valamint a különféle konfigurációs, strukturált adatmegjelenítést leíró fájlok. Az adatok fenti kritériumok szerinti ábrázolására az XML nyelv került kiválasztásra [5].

Implementálásra került egy modul, mely XML olvasó és író metódusokat tartalmaz, továbbá lehetőséget biztosít az XML társtechnológiának a hatékony kihasználására [6,7]. Ezek lehetnek a konkrét nyelvtan ellenőrzésére, vagy XSLT transzformációk elvégzésére a többszintű leírás kialakítására. A többszintű adatleírás szükségességét az ember szempontjából való jól olvashatóság és a gép számára könnyen feldolgozhatóság közötti esetleges hézag, illetve a rugalmas felhasználás indokolja. Ennek köszönhetően a modul forráskódja nagymértékben független lesz a felhasználási környezettől, csak a szükséges transzformációkat tartalmazó fájlokat kell megadni.

#### 4. Összefoglalás

A cikkben bemutatásra került, hogy az űrkutatásban használatos földi ellenőrzőberendezések fejlesztése kapcsán milyen programozástechnikai kérdések merülnek fel, és hogy ezek a kérdések miként lettek feloldva a területen szerzett sokéves tapasztalatnak köszönhetően. Írásunk első része az EGSE rendszerek általános jellemzését, majd a KFKI RMKI és az SGF Kft. jelenlegi fejlesztési munkáit foglalta össze.

#### Köszönetnyilvánítás

A bemutatott eredmények a Magyar Űrkutatási Iroda, az ESA PECS irodája valamint az MTA támogatása alapján jöhettek létre.

#### A szerzőkről

**Balajthy Kálmán** 1975-ben villamosmérnöki, 1988-ban gazdasági mérnöki diplomát szerzett a BME-n. 17 éven át a KFKI MSzKI-ban számítógépek ipari alkalmazásával foglalkozott. Szakterülete a valós idejű, idő- és helykritikus programok, nagyméretű alkalmazások elkészítése volt. Ezt követően egy német-magyar cégnél 9 éven át energia-elszámolási és számlázási programok fejlesztését végző programozói csoportot vezetett. 2004 óta a KFKI RMKI-ban űrkutatási projektek beágyazott processzorainak programfejlesztésében vesz részt.

**Lipusz Csaba** 1986-ban végzett fizikus szakon az ELTE TTK-n. A hardverközelű programozás és az informatikai rendszerek technikai architektúrájának fejlesztése azok a területek, melyeken pályafutása során több évet dolgozott. Jelenleg fedélzeti adatgyűjtő berendezések real-time Linux operációs rendszereivel és illesztő kártyák device drivereinek fejlesztésével foglalkozik.

**Sódor Bálint** a BME Villamosmérnöki és Informatikai Karán az informatikai infrastruktúrák tervezése szakirányon szerzett műszaki informatikusi végzettséget 2006-ban. Egyetemi tanulmányai során került kapcsolatba a KFKI RMKI-val, ahol missziókritikus rendszerek tesztelésével és szimulációjával foglalkozott és itt is került főállásba. Jelenleg az űrmissziók műszereihez alkalmazott tesztelő és szimulációs eszközök szoftverfejlesztésén dolgozik.

**Szalai Sándor** 1962-ben villamosmérnöki, 1967-ben folyamatszabályozási szakmérnöki oklevelet szerzett a BME-n. Kezdetben a nukleáris elektronika, majd 1980-tól az űrkutatás területén fejlesztett vezérlő- és adatgyűjtő-rendszereket. Az első munkái hardver-, majd később hardverközelű szoftver-fejlesztések, valamint rendszertervezés voltak. 1977-ben kandidátus, 1992-ben a műszaki tudomány doktora lett. 1977-től a KFKI RMKI-ban dolgozik, jelenleg tudományos tanácsadó. 1996-tól az SGF Kft. ügyvezető igazgatója.

#### Irodalom

- [1] <http://www.w3.org/MarkUp/SGML/>
- [2] <http://www.w3.org/MarkUp/>
- [3] Balajthy K., Sulyán J., Szalai L., Sódor B., Lipusz Cs., dr. Szalai S., Elosztott intelligenciájú automatizált rendszer a VenusExpress űrmisszió kísérletének kalibrálására. *Híradástechnika*, 2006/04, pp.34–38.
- [4] Balajthy K., Endrőczy G. dr Nagy J., Horváth I., Lipusz Cs., dr. Szalai S., Adatgyűjtő és vezérlő számítógép a Nemzetközi Űrállomás Obsztanovka kísérletéhez. *Híradástechnika*, 2006/04, pp.17–22.
- [5] Brett McLaughlin, *Java és XML*. O'Reilly, Kossuth, 2001, ISBN: 963 09 4230 5.
- [6] B. Sódor, G. Tróznai, Cs. Lipusz, *Implementing Data Presentation Layer in Testing and Simulation Environments Using XML*. 58th Int. Astronautical Congress Hyderabad, India, 24-28 September 2007, IAC-07-D1.1.11.
- [7] Sódor Bálint, *XML használata tesztelő, szimulációs rendszerekben*. Elektronet, 2006. december, pp.12–14.