A KOMPLEX RENDSZEREK KOMMUNIKÁCIÓ VEZÉRELT TERVEZÉSE

Ady László[1], Tokody Dániel[1], Mester Gyula[1], Hudasi Luca Flóra[1]

[1] NextTechnologies Komplex Rendszerek Kutató Intézet, Maglód

KIVONAT

A komplex rendszerek egyre elterjedtebbek. Szinte minden modern rendszer elosztott. A kiadási ciklusok rövidülésével a hibák, balesetek kockázata növekszik. Ez a kockázat a rendszerek komplex rendszerként kezelt tervezése által csökkenthető. A komplex rendszerekben a kommunikáció az egyik legjellemzőbb leírása a kollektív viselkedésnek. Komplex rendszerek tervezése két fő részre bontható, a részegységek viselkedése és a részegységek együttműködése, kommunikációja. Ezeket a protokoll írja le. A protokoll egyik jellemző feladata a részegységek hibájának és a protokoll belső hibájának eszkalációját gátolni. Komplex rendszerek tervezési módszereit tekintjük át és foglaljuk össze a kommunikáció, protokollok szempontjából.

Kulcsszavak: komplex rendszerek, kommunikáció, protokoll, tervezés.

1. BEVEZETŐ

A mai és a jövő rendszerei jellemzően hálózatba kötöttek. Ezek a hálózatok jellemzően eseti (ad hoc) hálózatok. A hálózatra kapcsolt különböző egyedek együttműködése eredményezi a rendszer kívánt működését. A rendszer által nyújtott elvárt szolgáltatások minőségi és mennyiségi paraméterei általánosan nagyon eltérőek lehetnek. Ebből fakadóan az elvárások is széles skálán mozognak. Ebből következik, hogy egy ilyen rendszer tervezéséhez számos különböző lehetőség adódik és rendszerenként eltérő, hogy melyik paraméterek szerint kell kiválasztani ezeket.

A rendszerek tervezésénél ezért szükséges a paramétereket és a rendszer kialakítását előzetesen meghatározni (Tokody 2020). A rendszer kialakítása lehet:

- általános,
- biztonságkritikus (safety-critical):
    - hiba esetén is működő (fail-operational),
    - hiba esetén csökkentet üzemmód (fail-soft),

- hibabiztos (fail-safe),
- hiba biztonság(fail-secure),
- hibára nem reagáló (fail-passive),
- hibatűrő (fail-tolerant).

A biztonságkritikus rendszer életciklusa domainenként különböző előírások szerint történik. Dominánsan ezek szabványok formájában jelennek meg. Jellemzően ezek a IEC61508-ra épülnek. Ez a műszaki eszközöket biztonsági szempontból osztályokba sorolja és meghatározza a szintekhez szükséges eljárásokat az életciklus során (Schuster 2018).

Főbb biztonságkritikus szabványok:

- EN 50126, EN 50128, EN 50129 vasúti környezetben
- DO-178C avionika
- ANSI/ISA S84, IEC 61511 ipari folyamatirányítás
- IEC 61513 nukleáris ipar
- IEC 62061 gépbiztonság
- ISO 26262 járműipar

Jellemzően a mai és a jövőben kialakított komplex műszaki rendszerek integráltak. Például:

- Smart City,
- ITS,
- épületautomatikák és ezek hálózata,
- kooperatív robotok hálózata

Legtöbb ilyen rendszertől probléma terének mérete lehetetlenné teszi az egzakt vezérlő rendszerek alkalmazását, ezért valamilyen heurisztikát vagy lágy számítási megoldást tartalmaz. A biztonságkritikus rendszerek fejlesztése során elvárás a determinisztikus viselkedés. Öntanuló lágy számítási rendszerekkel a determinisztikus viselkedés nehezen megvalósítható. Ezt az ellentmondást a rendszerek tervezése során úgy kell feloldani, hogy az elfogadható kockázat szintje alá csökkenjen a rendszer maradó kockázata.

A dolgozat célja feltárni a komplex rendszerek tervezésének azon módját amikor a rendszer külső és belső kommunikációja határozza meg az optimális tervezés fő vezérlő elvét.

2. KOMPLEX RENDSZER

2.1. Komplexitás

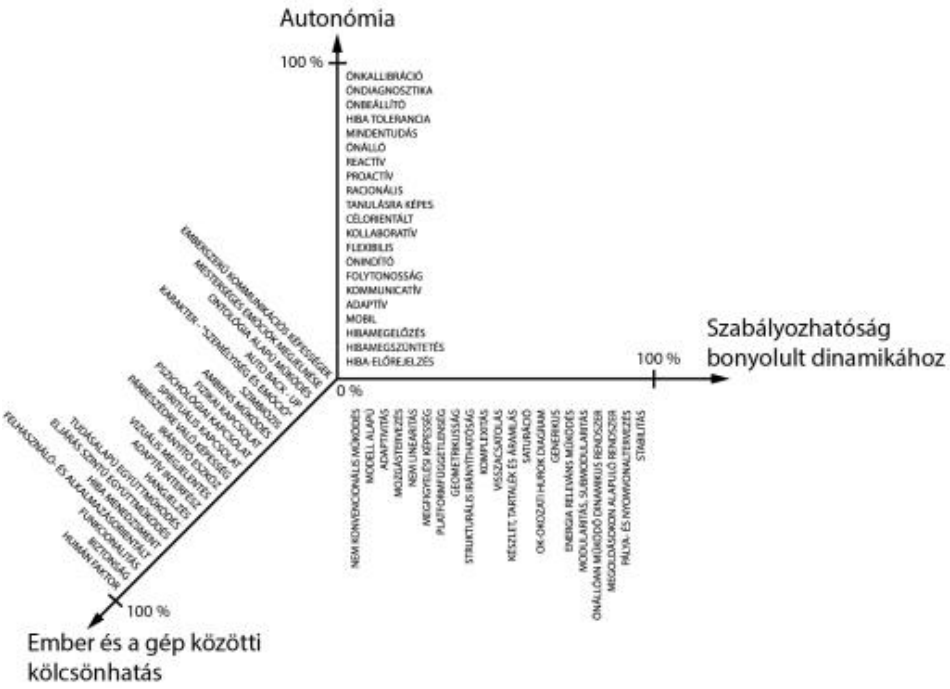
A rendszerek komplexitása többféle okból állhat elő, mint például:

- méret nagyságából fakadó,
- a megoldandó problématér méretéből fakadó,
- ön szerveződésből fakadó,
- nem lineáris viselkedésből fakadó.

Egy önállóan nem komplex eszköz nagyobb rendszerbe illesztéssel előállhat olyan helyzet amikor a kollektív viselkedés megbecslése vagy leírása már nehezen megadható. Például egy UAV fedélzeti irányitó rendszerek közepes bonyolultságú, leírható (Mester 2015), (Mester 2013). Ugyan akkor lehet alkotni lágy számítási módszerekkel leírt irányitó rendszert is UAV avionikákhoz (Nemes 2017). Viszont az UAV-t egy rendszerbe illesztve könnyen lehet olyan állapotot találni amikor a viselkedése kaotikussá válhat és az UAV-k együttműködése külön extra tervezést igényel. Például UAV-kal kontakt mérések végre hajtása, miközben a kontakt méréshez segédanyagot kell juttatni. Amikor a kontakt szenzor a felülethez ér (és egy méréshez szükséges nyomó erő biztositásra kerül) vagy a felülettől elválik az UAV irányítása kis időre instabil állapotba kerül, a másik UAV-nak vagy a közöttük lévő segédanyag ellátásnak ezt le kell tudnia kezelni.

2.2. Komplex rendszerek intelligenciája

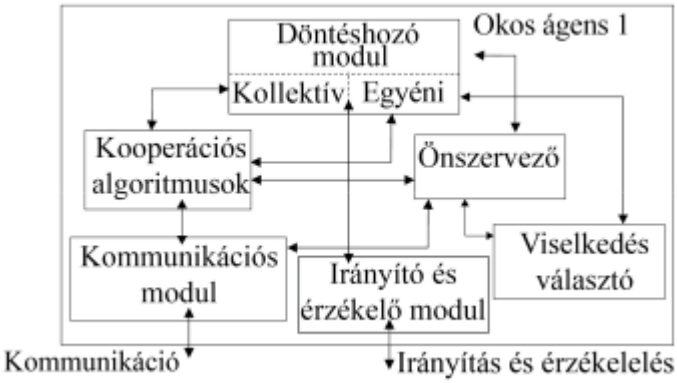
Az ilyen rendszerek vizsgálati alapja az általános rendszer elmélet tovább gondolásával lehetséges. Ezt Tokody Dániel a doktori disszertációjában végig vitte (Tokody 2020). A munkásságát felhasználva a komplex rendszerek intelligenciája merhetővé vált.



1. ábra. Az MIQ összetevőinek ábrázolása (Tokody 2020)

2.3. Elosztott komplex rendszerek ágens alapú megközelítése

Az elosztott komplex rendszerek álhatnak statikus, dinamikus és intelligens elmékből. Ezeket tervezés szempontjából célszerű külön kezelni. Intelligens ágens felépítéséből fakadóan önállóan komplex rendszerként kezelhető míg a statikus, dinamikus ágens egyszerű tervezési módszerekkel is kezelhető. A közöttük fellépő interakciót viszont komplex rendszerként kell mindig kezelni.



2. ábra. Okos ágens felépítése (Tokody 2020)

2.4. Komplex rendszerek hálózata

Komplex rendszerekben kiemelt fontosságú, hogy különálló elemek kommunikációja alakítja ki a kollektív viselkedést. Az elemek hálózatba kapcsolva kommunikálnak. A hálózat hozzáférés szempontjából lehet:

- I osztályú,
- II osztályú,
- III osztályú.

1. táblázat. Hálózati besorolás EN 50129 szerint (saját szerkesztés)

Tulajdonság	I osztályú	II osztályú	III osztályú
Csatlakoztatható eszközök	Csak a gyártó által engedélyezett	Idegen eszközök	Idegen eszközök
Eszközök száma	Rögzített	Változhat	Változhat
Hálózat beállítása	Előírt	Változhat	Változhat
Illatéktelen hozzáférés	Elhanyagolható	Elhanyagolható	Nem elhanyagolható

2.4.1. Hálózati zavarok

A valós hálózattokon kommunikációs zavarokkal kell számolni ezek lehetnek:

- Üzenetismétlődés,
- üzenet-kimaradás,
- üzenet beékelődés,
- üzenet újra számozás,
- adatkorrupció,
- üzenetkésés,
- üzenet maszkolás,
- résztvevők beállítás hiba,
- FIFO-hiba.

2.5. Okos komplex rendszerek tervezése

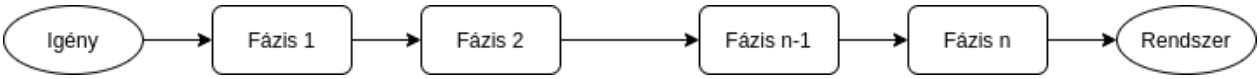
2.5.1. Tervezési módszerek osztályozása

A tervezési módszereket az életciklus fázisai szerint a fázisok egymásutánisága és fázisok ismételhetősége szerint csoportosítva:

- lineáris,
- spirális,

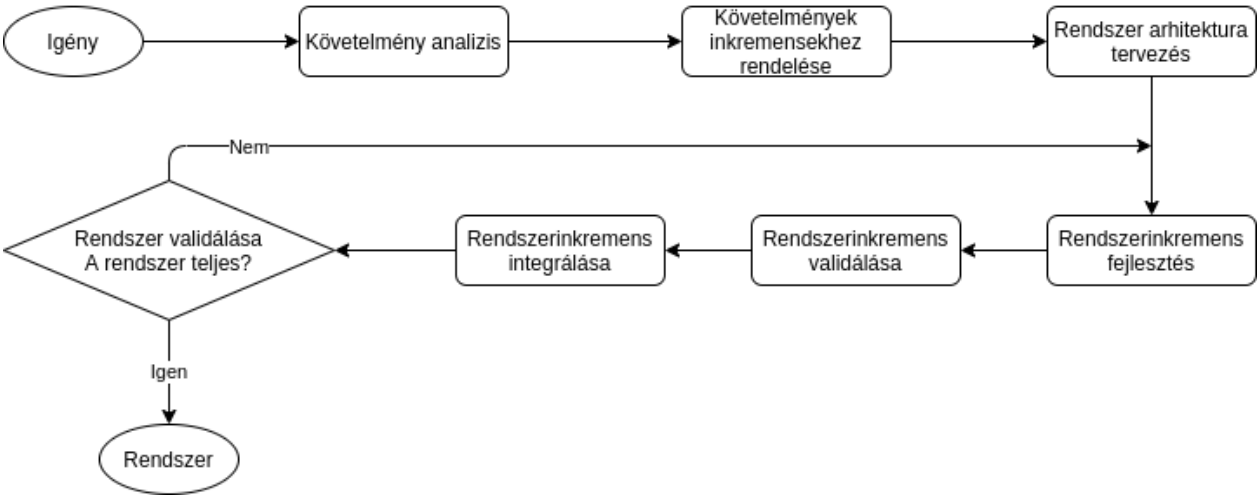
- iteratív.

Az lineáris fejlesztés során a fázisok egymást követik. Jellemzően olyan rendszerekhez alkalmas, ahol a környezet nem változik, egyes fázisok nem megismételhetőek.



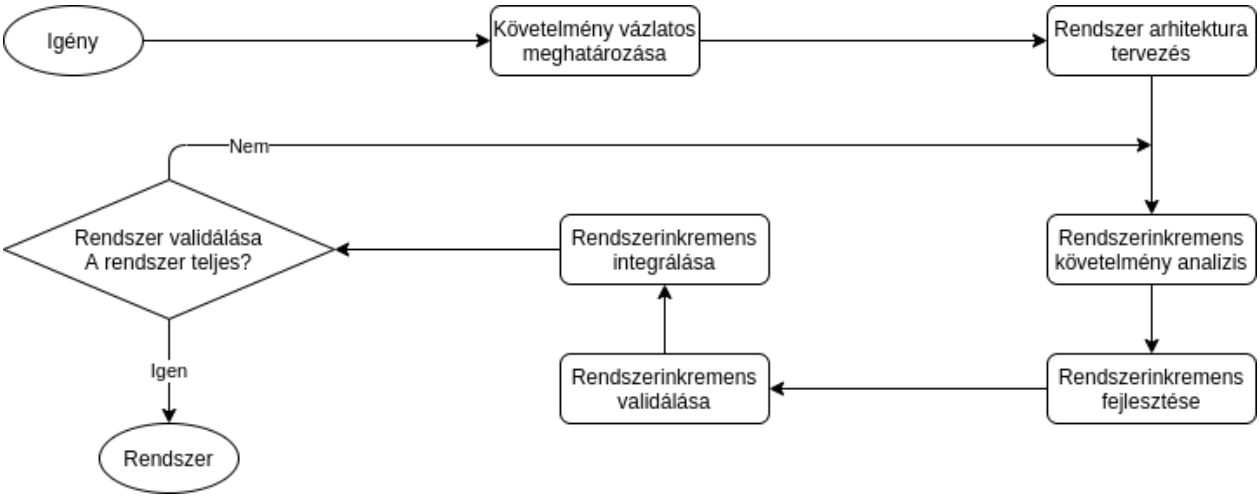
3. ábra. Lineáris fejlesztés

Spirális fejlesztés során a fázisok finomodva ismétlik egymást. Boehm 1986-ban írta le. A fejlesztés során a folyamat egy-egy fázisai közötti kapcsolatot spirálként reprezentálja. Vagyis minden fejlesztési körben ugyan azok a folyamatok játszódnak le a körben meghatározott célok elérése érdekében. A kört inkrementálásnak nevezik. Az elérendő célt inkremensnek nevezik és a kör végére az inkremens integrálásra és validálásra kerül.



4. ábra. Spirális fejlesztés (saját szerkesztés)

Iteratív folyamat hasonló a spirális fejlesztéssel azzal a kivétellel, hogy a specifikációt a szoftverrel összekapcsolva kell fejleszteni, nem pedig előre elkészíteni az egész dokumentumot.



5. ábra. Iteratív fejlesztés (saját szerkesztés)

A tervezési módszereket az életciklus adott domainre jellemző elvárt dokumentáció igénye szerint csoportosítva:

- könnyűsúlyú,
- nehézsúlyú

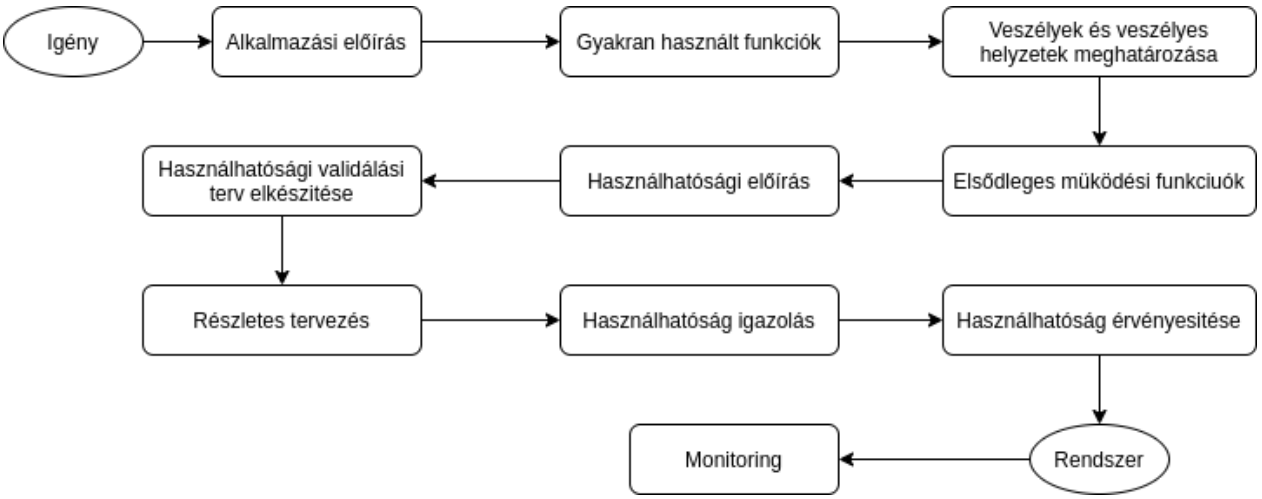
A könnyűsúlyú dokumentáció alacsony dokumentáltságot vár el és csak az általános kialakítású rendszerek fejlesztéséhez lehet alkalmazni.

A nehézsúlyú dokumentáció doménenként változó, azonmban jellemző, hogy mint mennyiségre, minőségre, formátumra és kezelésére merev megkötések vannak.

A tervezési módszereket az életciklus során előtérbe helyezett módszertan szerint csoportosítva:

- adatközpontú,
- folyamatközpontú,
- követelményközpontú,
- használatieset-központú,
- tesztközpontú,
- felhasználóközpontú,
- emberközpontú,
- csapatközpontú.

Okos komplex rendszerek tervezése az általános használhatósági tervezésből származtatható. Ez egy jól kiforrott szabványosított folyamat. Részletesen megtekinthető a ‘Intelligens vasúti informatikai és biztonsági rendszerek fejlesztése’ disszertációban.



6. ábra. Okos rendszerek használhatósági tervezése ‘Intelligens vasúti informatikai és biztonsági rendszerek fejlesztése’ szerint (saját szerkesztés)

3. TERVEZÉS

3.1. Előkészítés

A modern komplex rendszerek tervezésénél figyelembe kell venni, hogy ilyen rendszert várhatóan nagy költségen és hosszú üzemre kell tervezni és az életciklusa alatt több átalakítás várható, miközben nyílt rendszerként kell kezelni. Vagyis III. Osztályú hálózat, és valamilyen ismétlődő fejlesztési ciklust kell alkalmazni. A rendszer tervezés elején a rendszer felhasználási jellegének megfelelő módon kockázatot kell elemezni és meg kell határozni a rendszer biztonsági besorolását. Ennek megfelelően a felhasználható módszertan:

- agilis,
- SafeScrum (Hansen,Stalhane,Myklebust 2018).

3.2. A rendszer környezete és a rendszer kollektív viselkedésének leírása

A rendszer és a környezete közötti kommunikáció meghatározása:

- adatátvitel kezdeményezése és befejezése,
- küldők és fogadók szinkronizálása,
- küldési hibák észlelése és kijavítása,
- adatok formázása és kódolása,
- üzenetek számossága,
- tesztelhetőség, teszt leírások,

- üzenetek prioritása.

A rendszer és a környezete között az elvárt viselkedést, kommunikációt keretek közé kell szorítani. Ezeket a keretek a kommunikációs protokollok.

A környezet megkötései, amiben protokoll létezik:

- legalább két kommunikációban résztvevő,
- legalább egy adatküldési csatorna.

3.3. Protokoll felépítési módszerek

A protokoll felépítési módszerek az elemi jel felépítésére utalnak. Ezek lehetnek fizikai vagy információ technológiai elemek.

Fizikai például:

- egyenfeszültségű távadó 0-10V,0-5V,2-10V,
- egyenáramú távadó 0-20mA,4-20mA,
- villamos áram frekvenciája,
- rádió hullám,
- mágneses mező,
- optikai,
- mechanikai erő, nyomaték,
- pneumatika,
- hidraulika,
- Kémiai.

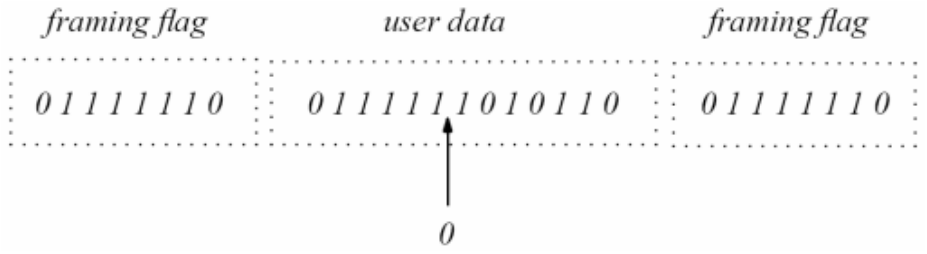
Infomráció technológiai például:

- bit orientált,
- karakter orientált,
- byte-számolás orientált.

A bit orientált protokollok egyidőben a csatorna bitszámával megegyező bitet küld. Ilyen például a számitógépek párhuzamos portja (LPT port). A biteket időben el kell választani hogy a fogadó tudja, hogy mikor ér véget az egyik bit és hol kezdődik a másik bit. Ez lehet a:

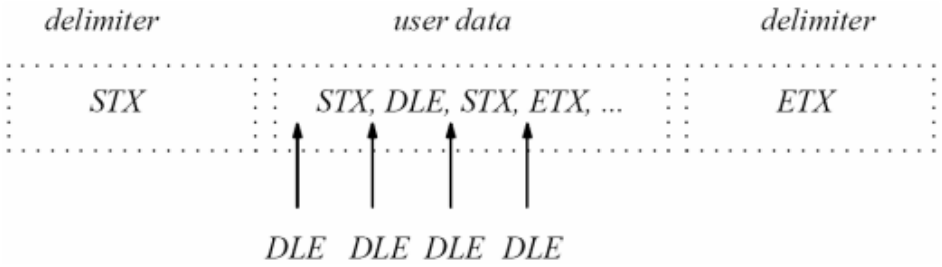
- felhasználó által küldött adat része,
- keret része,
- külön álló flag vagy órajel.

Ha a felhasználó által küldött adatban a keret vagy tiltott bit kombináció van akkor a felhasználó által küldendő mintát fel lehet tördelni különálló darabokra.



7. ábra. Bit orientált keret (Holzmann)

A karakter orientált protokollban jellemzően kis adat struktúrák kerülnek küldésre, amelyek karakterekből állnak (általában 7 vagy 8 bit). És az üzenet mindig ezek n többszörösei a karakter méretének. A felhasználó által küldött vezérlő karakterekkel kell foglalkozni. Amennyiben ilyen fordul elő akkor ki lehet kapcsolni a vezérlő karakter feldolgozását a DLE (data link escape) segítségével.



8. ábra. Karakter orientált keret (Holzmann)

Byte számolás orientált protokoll esetén a fejlécben megadásra kerül az üzenet mérete igy az üzenet elválasztás ez alapján működik. A mai protokollok többsége ezt használja. Azonban ott ahol kiemelt megbízhatóságra van szükség célszerű fix üzenet hosszúságú protokollt kialakítani.

Valós rendszerek esetén számolni és tervezni kell zavarokkal és meghibásodásokkal.

Protokoll leírás tartalmazza a:

- protokoll szótárát (üzenettípusok),
- üzenet formátum leírását,
- eljárási szabályokat,
- feltételezéseket a környezetről,

- prioritások,
- üzenetek várható mennyisége, mérete,
- protokoll által nyújtott szolgáltatás leírása.

A protokollokat formálisan le kell írni és elemezni kell. Erre alkalmas módszerek:

- a formális módszerek,
- a Petri hálók

Formális módszerek:

- CSP (Communicating Sequential Processes),
- CCS (Calculus of Communicating Systems),
- HOL (Higher Order Logic),
- LOTUS (Language for Temporal Ordering Specification),
- OBJ,
- Temporal logic,
- VDM (Vienna Development Method),
- Z módszer,
- B módszer,
- Model Checking.

A formálisan leírt kommunikáción matematikai elemzéseket lehet végre hajtani. Továbbá lehetővé válik:

- protokoll formális modellezése és analízise;
- követelmények formalizálása;
- prokoll formális verifikációja modellellenőrzéssel;
- állapotfüggő dinamikus viselkedés modellezése;
- konkurens rendszerek modellezése és analízise;
- adatfüggő viselkedés modellezése;
- adatfeldolgozás modellezése;
- rendszer terv (vagy szoftver esetén forráskod) alapú formális verifikációs technikák.

3.2. Protokoll tervezési szabályok

A protokoll tervezése során figyelembe veendő alapszabályok:

- legyen egyszerű,
- legyen modularitás,
- legyen jól formált,
- legyen robosztus,
- legyen konzisztens.

Ezen kívül a biztonság kritikus környezetben a domain specifikus szabályok előírására is szükség lehet, ami kiterjedhet:

- csatornák számára,
- időzítésekre,
- titkosításokra,
- hiba detektálásra,
- hiba javításra.

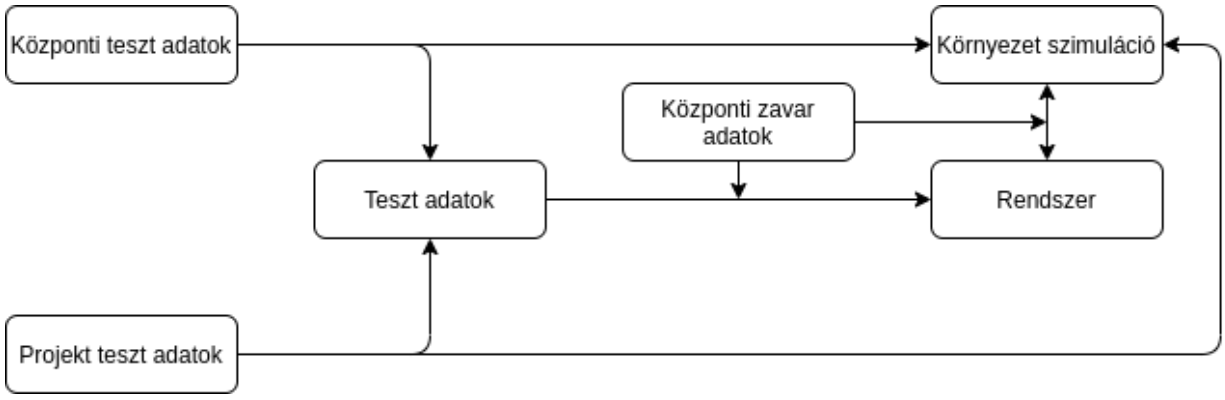
3.4. Protokoll tervezési hibák

Lehetséges hibák:

- nem veszi figyelembe a protokoll tervezése során figyelembe vehendő alap szabályokat,
- kommunikációt megzavarják a zavarok nincs figyelembe véve a komplex rendszereken előforduló lehetséges zavaró hatások,
- hibáson kerül felmérésre a környezet,
- hibásan kerül besorolásra biztonság kritikusság szempontjából,
- tesztelési tervek nem megfelelően fedik le a hiba lehetőségeket.

3.5. Tesztelés

A formális specifikáció és leírás lehetőséget teremt a zavaró hatások formális leírására. Ezek a zavarok más rendszerekre is igazak és felhasználhatóak. Vagyis létre lehet hozni zavar és teszt adatbázist, ami jól használható több projektben is.



9. ábra. Automata teszt környezet (saját szerkesztés)

4. ZÁRÓ KÖVETKEZTETÉSEK

A tervezés korai fázisban végzett formális leírással létrehozott specifikáció megköveteli a részletes átgondolást így a rendszer működésének a korai megértését. A formális eszközök átláthatóvá teszik a rendszerrel szemben támasztott követelményeket, rendszer viselkedését. A központi formális leírások egy olyan mankót adnak, ami segítségével lehet látni, hogy a formális leírás vagy a specifikáció menyire tekinthető teljesnek.

A formális leírás és a formális leírás által végezhető elemzések, generálások nagy mértékben csökkentik a hiba arányt. A modulálásán be kapcsolható központi teszt adatok és zavar jelek segítségével a rendszer szimulált környezetben tesztelhető gyorsan a leg különfélébb környezet és rendszer állapot együtt állasokkal.

Ez összeségben idő megtartást jelent és garantál egyfajta stukturáltságot.

A jövőben szeretnénk több projekten kipróbálni és a központi teszt és központi zavar adatbázist teljessé tenni. Tovább automatizálni a folyamatokat és egy teljes részletes leírást készíteni, amivel az összes ismert hiba lehetőség elkerülhető. (Például minden üzenetnek legyen GUID-je és minden üzenet legyen idempotens)

IRODALOMJEGYZÉK

Tokody Dániel. 2020. Intelligens vasúti informatikai és biztonsági rendszerek fejlesztése (Doktori értekezés). Óbudai Egyetem, Biztonságtudományi Doktori Iskola, Budapest. http://lib.uni-obuda.hu/sites/lib.uni-obuda.hu/files/Tokody_Daniel_ertekezes.pdf (2020.10.10).

Schuster György, Ady László. 2018. Biztonságkritikus szoftver fejlesztés. Repüléstudományi közlemények XXX. évfolyam https://www.researchgate.net/publication/331718540_BIZTONSAGKRITIKUS_SZOFTVER_FEJLESZTES__SAFETY_CRITICAL_SOFTWARE_DEVELOPMENT (2020.10.10)

Nemes Atilla, Mester Gyula. 2017. Unconstrained Evolutionary and Gradient Descent-Based Tuning of Fuzzy-partitions for UAV Dynamic Modeling. FME Transactions, Vol. 45, No. 1, pp. 1-8. DOI:10.5937/fmet1701001N.

Mester Gyula. 2015. Modeling of Autonomous Hexa-Rotor Microcopter. Proceedings of the IIIrd International Conference and Workshop Mechatronics in Practice and Education (MechEdu 2015), pp. 88-91, ISBN 978-86-918815-0-4, Subotica, Serbia.

Mester Gyula. 2015. Backstepping Control for Hexa-Rotor Microcopter. Acta Technica Corviniensis – Bulletin of Engineering, Vol. 8, No.3, pp. 121-125.

Mester Gyula, Rodic A. 2013 Simulation of Quad-rotor Flight Dynamics for the Analysis of Control, Spatial Navigation and Obstacle Avoidance. Proceedings of the 3rd International Workshop on Advanced Computational Intelligence and Intelligent Informatics (IWACIII 2013), pp. 1-4, ISSN: 2185-758X, Shanghai, China.

Mester Gyula, Rodic A. 2012. Navigation of an Autonomous Outdoor Quadrotor Helicopter. Proceedings of the 2nd International Conference on Internet Society Technologie and Management ICIST, pp. 259-262.

Geir Kjetil Hanssen, Tor Stålhane, Thor Myklebust. 2018. SafeScrum ® – Agile Development of Safety-Critical Software.

Ady László, Tokody Dániel. 2019. Komplex rendszerek kommunikációjának hatásai és tervezési irányelvei.