Acta Agraria Kaposváriensis (2004) Vol 8 No 3, 99-116
Kaposvári Egyetem, Állattudományi Kar, Kaposvár
University of Kaposvár, Faculty of Animal Science, Kaposvár

# Állapotmonitorozási és megbízhatósági információk integrálása erőművi informatikai rendszerekben

**Szabó G., [1]Varga I., [1]Bartha T.**
Budapesti Műszaki és Gazdaságtudományi Egyetem, Közlekedésautomatikai Tanszék
Budapest, 1111 Bertalan L. u. 2.
[1]Magyar Tudományos Akadémia Számítástechnikai és Automatizálási Kutató Intézete
Rendszer- és Irányításelméleti Kutató Labor, Budapest, 1111 Kende u 13-17.

## *ÖSSZEFOGLALÁS*

*Az erőművi informatikai rendszerek alkalmazásának egyik célja az erőmű állapotának folyamatos monitorozása. A biztonságkritikus területen alkalmazott alrendszerek minőségének egyik legfontosabb aspektusa a megbízhatóság. Az ilyen rendszerek tervezése során a megbízhatósági paraméterek teljesítése az egyik legfontosabb tervezési szempont, ezért a paraméterek számszerűsítésre is kerülnek. Az erőművi rendszerek korszerűsödésével egyre több státuszinformáció áll rendelkezésre, az egyes részrendszerek hibás vagy degradálódott állapotáról. A státuszinformációt feldolgozva, majd a hibamodellbe integrálva olyan hibamodell nyerhető, amely a rendszer aktuális állapotát követi. A modell kiértékelése által számszerűen meghatározható a rendszer degradációjának foka. Ez a degradációs mutató fontos tájékoztatást jelent a bekövetkezett hibaesemény(ek) súlyára vonatkozólag.*
(Kulcsszavak: diagnosztika, produkciós rendszer, erőművi megvalósítás, hibafa, megbízhatósági analízis)

## *ABSTRACT*

**Integration of status monitoring and reliability analysis in the information systems of power plants**

G. Szabó, I. [1]Varga, T. [1]Bartha
Budapest University of Technology and Economics, Department of Transport Automation
Budapest, H-1111 Bertalan L. u. 2.
[1]Systems and Control Laboratory, Computer and Automation Research Institute Hungarian Academy of Sciences
Budapest, H-1111 Kende u. 13-17.

*Among the reasons for the utilisation of computer engineering in power plants is the continuous monitoring of the plant's state. In the case of subsystems used in the area of critical safety one of the most important aspects of quality is the reliability of the systems. When designing such systems the realisation of safety parameters is essential, therefore these parameters are numerically expressed. Along with the modernisation of the power plants systems, the availability of state information about certain subsystems' defective status is on the increase. Processing the status information and integrating it into the error-model, a model that follows the system's actual state, can be obtained. By evaluating the model, the degree of the system's degradation can be numerically determined. This degradation indicator gives important information about the gravity of the error-event(s) that occur.*
(Keywords: diagnostics, production system, realisation in power plant, fault tree, reliability analysis)

## BEVEZETÉS

A biztonságkritikus környezetbe telepített védelmi rendszerek esetlegesen bekövetkező hibák esetén is tovább működnek, hiszen szinte kizárólag hibatűrő felépítésűek. A bekövetkezett hiba hatására azonban a rendszer képességei megváltoznak. Ezekben az esetekben a lehető leggyorsabban fel kell tárni a hibát, és meg kell határozni annak hatását a tulajdonságokra (elsősorban a védelmi képességekre), hogy azok mennyiben változtak, degradálódtak. A hiba bekövetkezése után két feladatot kell elvégezni:

– hibalokalizálás,
– degradáció-elemzés.

A két funkciót egyesítő „Hibalokalizáló és degradáció-elemző modul" alkalmazásának célja a Paksi Atomerőmű új Reaktorvédelmi Rendszerében az *RVR állapot ellenőrző* rendszer által szolgáltatott adatok feldolgozása és az üzemeltető, továbbá technológusi személyzet jelenleginél pontosabb tájékoztatása az RVR-ben bekövetkezett meghibásodásokról. Másik fontos feladat a hibafa analízisen alapuló degradáció-elemzés, amelynek működése a hibalokalizáló modul által készített vizsgálat eredményeire támaszkodik.

A következő fejezetekben bemutatjuk a hibalokalizálás elvét a konkrét megvalósítást, majd ezen modul eredményeit felhasználva ismertetjük a degradáció elemzés lépéseit. A „Hibalokalizáló és degradáció-elemző modul" a Paksi Atomerőműben a COSMOS projekt részeként fejlesztették ki. **COSMOS=C**omputerised **O**peration **S**upport for **M**anagement, **O**ptimisation and **S**urveillance (Tudásintenzív információs technológia bonyolult ipari rendszerek biztonságos és optimális működtetéséhez).

## ANYAG ÉS MÓDSZER

### Hibalokalizálás diszkrét rendszerekben

A diszkrét rendszerekhez kidolgozott hibafelismerő és lokalizáló modul a digitális folyamatirányító számítógépeknek és számítógép hálózatoknak, mint önálló rendszereknek vizsgálatát célozza meg. Ezek a rendszerek a sokoldalú felhasználhatóság érdekében általános elvek szerint épülnek fel és kevéssé specifikusak az alkalmazási környezetre nézve. Ugyanakkor bonyolultságuk sokszor összevethető az általuk monitorozott, illetve szabályozott folyamatok és termelési rendszerek összetettségével. A bonyolult felépítés és a nagy alkatelem szám egyrészt megnöveli a meghibásodások lehetőségét, másrészt egy bekövetkezett hiba pontos okának felderítése esetenként komoly szakértői ismereteket és hosszas vizsgálatokat igényel. Célszerű tehát ezeket a rendszereket önálló, az ipari folyamattól jórészt független, saját hibafelismerő és tanácsadó számítógépes alrendszerrel ellátni.

### A hibalokalizálási és diagnosztikai feladat diszkrét rendszerekben

A hibafelismerést és lokalizálást végző modul az ipari rendszer berendezéseinek normálistól eltérő viselkedését mérések, tesztek és on-line számítások alapján korai fázisban detektáló eszköz. A modul mind a folyamat részét képező eszközök (mérő és beavatkozó szervek, valamint egyéb passzív és aktív komponensek), mind pedig a szabályozó/felügyelő (jellemzően digitális számítógép) rendszerek állapotának megfigyelésére képes. Egy diszkrét rendszer esetén a normális állapottól való eltérést (illetve annak lehetséges okait) a modul a komponensek hibamodellje és a rendszerben elvégzett tesztek eredményei alapján határozza meg. A hibafelismerés *modell-alapú*, ahol a formális modellt az adott rendszer elemeinek normális és hibás működés esetén tapasztalt viselkedési modellje és a hibamodell segítségével kidolgozott tesztek testesítik meg (*Fensel és Benjamins,* 1996).

A modell-alapú hibadiagnosztikai feladat formális leírása a modellezett rendszer leírásával, az ún. *diagnosztikai specifikációval* kezdődik (*Peter és Lucas,* 1998). A Σ diagnosztikai specifikáció egy olyan hármas $\Sigma=(\Delta,\Phi,e)$, ahol Δ jelöli a fizikai meghibásodásokat, illetve belső hibaállapotokat, Φ a rendszerben levő belső, illetve megfigyelhető hibajelzések halmaza és *e* a *tanúsítvány függvény*. A tanúsítvány függvény a hibás rendszer viselkedési leírása. Tartalmazza egyrészt a komponensek normális működésének leírását hibák jelenlétében (hibaterjedési jelenségek), a komponensek hibaszemantikáját, azaz viselkedését a komponens meghibásodása esetén, és a rendszer struktúráját, a komponensek közötti kapcsolatokat. A tanúsítvány függvény tehát egy

$$e : \wp(\Delta) \mapsto \wp(\Phi) \cup \{\perp\}$$

típusú leképezést hajt végre Δ és Φ hatványhalmazai között. Hibajelzés hiányában hibafelismerés és diagnosztika nem lehetséges, így elvárjuk, hogy minden $f\in\Phi$ hibajelzésre létezzen egy olyan meghibásodás halmaz, amelyet *f tanúsít*, azaz $e(D)=f$ vagy $e(D)=\neg f$, esetleg mindkettő. Ez utóbbi esetben az *f* hibajelzés léte vagy hiánya nem függ a *D* hibahalmaztól. A modell része a ⊥ szimbólummal jelölt ellentmondás vagy *inkonzisztencia* is, amely a diagnosztikai feladat megoldása során fontos szerephez jut. A tanúsítvány függvénnyel kapcsolatos elvárás, hogy minden lehetséges $D\subseteq\Delta$ meghibásodás halmazra, ha $d, \neg d \in D$ (tehát a ténybázis inkonzisztens), akkor a leírás ezt kimutassa, azaz $e(D)=\perp$. Ugyancsak fontos elvárt tulajdonság, hogy minden $D, D'\subseteq\Delta$ meghibásodás halmazra, ha $e(D)\neq\perp$, azaz ha az $e(D)$ hibajelzés halmaz *megfigyelhető*, akkor ez teljesüljön *D* minden részhalmazára is, tehát $D'\subseteq D$ esetén $e(D')\neq\perp$ is megfigyelhető legyen.

A diagnosztikai specifikáció leírja a rendszer hibamodelljét. A hibamodell ismeretében a *diagnosztikai probléma* $P=(\Sigma, E)$ a Σ specifikáció kiegészítése az $E\subseteq\Phi$ *megfigyelések* halmazával. A diagnosztikai probléma megoldását egy *diagnosztikai algoritmus* formájában keressük, amely egy

$$R_{\Sigma,e|H} : \wp(\Phi) \mapsto \wp(\Delta) \cup \{u\}$$

típusú leképezést valósít meg. A diagnosztikai algoritmust egy konkrét megfigyelésre alkalmazva áll elő az $R_{\Sigma,e|H}(E) = D$, $D\subseteq\Delta$ formájú *diagnosztikai eredmény*, azaz az algoritmus a megfigyelésekből következtet vissza a meghibásodások azon körére, amely azt előállíthatta. Az algoritmus kimenete lehet az *u* szimbólum is, amely azt jeleníti meg, hogy a diagnosztikai problémának az adott megfigyelések mellett nincs megoldása. Az algoritmusban használt $e_{|H}$ *korlátozott tanúsítvány függvény* azt fejezi ki, hogy a hibamodell méretét gyakorlati rendszerekben a modellméret kezelhető mértéken tartása érdekében általában valamilyen tervezési vagy tapasztalati szemponttal alátámasztott *H* hibahipotézisre korlátozva írjuk fel. Ilyenkor a modellezett fizikai meghibásodások, illetve belső hibaállapotok köre a $H\subseteq\Delta$ halmazra korlátozódik. Például egy kéthiba-tűrő, fail-stop rendszerben elég a diagnosztikai specifikációt az egy és kéthibás esetekre korlátozni, hiszen három, vagy annál több hiba bekövetkezése után a rendszer leáll, a hibajelzések elérhetetlenné válnak, azaz diagnosztika nem lehetséges.

A diagnosztika formális leírásából látható, hogy a diagnosztikai feladatban a tudásbázis szerepét a tanúsítvány függvény alakjában megadott viselkedési leírás játssza. Az *e(…)* tanúsítvány függvény központi szerepének köszönhetően a tulajdonságaiból számos diagnosztikai jellemzőre következtethetünk. Például amennyiben *e* monoton növekvő vagy csökkenő, akkor maga a diagnosztikai következtetési folyamat is *monoton*

lesz. Ha a tanúsítvány függvény invertálható, akkor az inverz tanúsítvány függvény segítségével olyan produkciós rendszer hozható létre, amely a diagnosztikai feladatot nem hátra, hanem előre irányuló következtetési mechanizmussal oldja meg. A tanúsítvány függvény lokális tulajdonságai alapján kategorizálhatók a *korrelált hibák*, a *hibaeltakarás* és egyéb diagnosztikai fogalmak. Az irodalomban (*Poole*, 1998a és 1998b) a diagnosztikai feladat megoldásának két fő stratégiája terjedt el.

1. *Konzisztencia-alapú diagnosztika*. A konzisztencia-alapú diagnosztika lényege, hogy a rendszermodellben a komponens hibaállapotok olyan hozzárendelését állítsa elő, amely konzisztens a megfigyelésekkel. A konzisztencia-alapú megközelítés a rendszermodell az elsőrendű predikátumkalkulus eszközeivel írja le. Három fő elemből áll: az **SD** halmaz predikátumokból alkotott kifejezésekkel a rendszer normális működését írja le, a **COMPS** halmaz a rendszerkomponenseknek megfeleltetett konstansok halmaza, az **OBS** halmaz pedig a megfigyeléseket leíró predikátum kifejezések halmaza. A diagnosztikai probléma megoldása olyan pozitív (a komponens hibás) vagy negatív (a komponens hibátlan) literálok rendelése a komponensek teljes (vagy részleges) köréhez, melyben a következő két feltétel teljesül:

   $$D = \{\text{Abnormal}(c) \mid c \in C\} \cup \{\neg\,\text{Abnormal}(c) \mid c \in \mathbf{COMPS} \setminus C\}$$

   ahol :

   $$C \subseteq \mathbf{COMPS}$$

   $$\mathbf{SD} \cup \mathbf{OBS} \cup D \not\Rightarrow \bot$$

   A bevezetőben használt jelölésrendszerrel a konzisztencia-alapú diagnosztika a következő módon definiálható:

   $$CB_{\Sigma,e|H}(E) = \begin{cases} H & \text{ha } \forall f \in E : f \in e_{|H}(H) \vee \neg f \notin e_{|H}(H), \\ u & \text{egyébként.} \end{cases}$$

2. *Kauzalitás-alapú (abduktív) diagnosztika*. A kauzalitás alapú diagnosztika a rendszer normális vagy hibás működését leíró modellt oksági kapcsolatok formájában írja le. Kétfajta oksági kapcsolat különböztethető meg: az *erős* és a *gyenge* kauzalitás. Erős kauzalitás esetén amennyiben egy adott meghibásodási halmaz jelen van a rendszerben, akkor a kauzálisan hozzárendelt hibajelzés halmaznak is jelen kell lennie. Ebben az esetben a tanúsítvány függvény kifejezhető logikai implikációk formájában a következő módon: $d_1 \wedge \ldots \wedge d_i \rightarrow d_j$ és $d_1 \wedge \ldots \wedge d_i \rightarrow f_k$ azaz a $d_1,\ldots,d_i$ meghibásodások együttes fellépte okozza a $d_j$ belső hibaállapotot és létrehozza az $f_k$ hibajelzést. Az oksági specifikáció $C=(\Delta,\Phi,\Re)$ a rendszer leírása ezekkel az eszközökkel. Elemei a meghibásodásokat jelölő literálokat tartalmazó $\Delta$ halmaz, a hibajelzésekhez rendelt $\Phi$ literál halmaz és a fenti logikai kifejezésekből álló $\Re$ viselkedési leírás. (Gyakori eset, hogy $\Re$ csak a rendszer hibás működésének leírására szorítkozik, ekkor az elemeit *abnormitási axiómáknak* szokás nevezni.) Amennyiben rendelkezésre áll a rendszermodell oksági specifikációja és az $E$ megfigyelt hibajelzések halmaza, az abduktív diagnosztikai feladat megoldása egy olyan $H \subseteq \Delta$ meghibásodás halmaz, amelyre az alábbi két feltétel teljesül:

   $$\forall f \in E : \Re \cup H \Rightarrow f,$$

   $$\forall f \in E^C : \Re \cup H \not\Rightarrow \neg f, \text{ ahol } E^C = \{\neg f \in \Phi \mid f \notin E\}$$

A két feltétel azt felezi ki, hogy a *H* halmazba foglalt hibahipotézisek igazoltnak tekinthetők, amennyiben jelenlétük maga után vonja az összes megfigyelt aktív hibajelzés felléptét (*fedési kritérium*), ugyanakkor nem hoz létre egyetlen olyan hibajelzést sem, amely a megfigyelések szerint inaktív (*konzisztencia kritérium*). A fenti erősen kauzális diagnosztikai problémáról könnyen igazolható, hogy nem monoton és diagnosztikailag nem független.

A gyenge kauzalitás elve azt mondja ki, hogy adott meghibásodási halmaz jelenléte nem feltétlenül vonja maga után a kauzálisan hozzárendelt hibajelzés halmaz minden elemének jelenlétét. Legjellemzőbb példa a gyenge kauzalitás elvének alkalmazására az intermittens hibák modellezése. Az intermittens hibák a rendszerben jelen levő olyan fizikai meghibásodások, amelyek lappangó belső hibaállapotokat hoznak létre, de ezek a belső hibaállapotok igen bonyolult aktiválódási mechanizmusokon keresztül jutnak ki a rendszer megfigyelhető kimenetére. Így az általuk létrehozott hibajelzések ritkán, sztochasztikus jelleggel jelentkeznek és hibaszemantikai szempontból a tranziens hibákkal azonosak.

Az erősen kauzális diagnosztikai probléma leírásában használt oksági specifikáció kis kiegészítéssel a gyengén kauzális diagnosztikai probléma leírására is alkalmas. Az $\Re$ viselkedési leírás logikai kifejezéseibe egy további tagot veszünk fel, az $\alpha^d_j$ és $\alpha^f_k$ alakú *feltételes literálokat.* Ezekkel a kiegészítésekkel a gyenge kauzális oksági specifikációban a logikai implikációk a $d_1 \wedge \ldots \wedge d_i \alpha^d_j \wedge \rightarrow d_j$ és $d_1 \wedge \ldots \wedge d_i \wedge \alpha^f_k \rightarrow f_k$ alakot öltik.

**A gyakorlati megvalósítás szempontjai és módszerei**

A diszkrét rendszerekben a modell alapú hibadiagnosztikai feladat fenti összefoglalása jól mutatja, hogy a feladat reprezentációjában és megoldásában jól használhatók a kauzális leírások. Erre a reprezentációs formára épül a szakértői rendszerekben igen széles körben használt *produkciós rendszer* típusú felépítés. Ezek három fő komponensből állnak: a következtetési folyamat pillanatnyi állapotát tükröző *tényadatbázis*, a szakértői tudást oksági kapcsolatok formájában tartalmazó tudásbázis és a következtetési folyamatot irányító *vezérlési stratégia*. Ez utóbbi a monoton következtetési folyamat esetén egyszerűen a tudásbázist alkotó szabályok feltételrészére végzett mintaillesztést, az illeszkedő szabályok közül a következő végrehajtandó kiválasztását és a következmények hatására létrejövő új tényeknek a tényadatbázishoz adását jelenti. A monotonitásnak köszönhetően ilyenkor a ténybázis tartalma folyamatosan bővül, egy korábban végrehajtott szabály következményeit sohasem kell visszavonni. A valós rendszerekben azonban igen ritkán tartható a következtetési folyamat monotonitása.

*Nemmonoton következtetés* esetén egyes szabályok végrehajtásakor a létrejövő új tények a tényadatbázis aktuális tartalmával egyesítve megsérthetik a konzisztencia feltételeket. Ilyenkor a diagnosztikai következtetési folyamat ellentmondásra jutott, egy vagy több már végrehajtott szabály visszaléptetésére van szükség. A visszalépés során a tényadatbázis aktuális állapotát (a modell elemeit megjelenítő logikai változók aktuális értékkombinációját) ki kell zárni a további vizsgálatból, majd vissza kell állítani a tényadatbázisnak a szabály végrehajtása előtti állapotát. A visszalépéssel kiegészített vezérlési stratégia tehát azt határozza meg, hogy hogyan (mennyire hatékonyan) járja be a következtetési folyamat a diagnosztikai modell állapotterét. A keresés hatékonysága feladatspecifikus heurisztikák bevonásával javítható. Az általánosan használható neminformált keresési módszerek hatékonyság szempontjából elmaradnak a heurisztikus módszerekkel szemben, azonban nagy előnyük, hogy nem tartalmaznak alkalmazásfüggő elemeket.

A produkciós rendszer alapú felépítés a Paksi Atomerőmű Reaktorvédelmi Rendszer hibadiagnosztikájának tervezett tartamát tekintve a feladathoz jól illeszkedő megoldásnak tekinthető. Számos előnye közül néhány: széles körű irodalmi és alkalmazási háttérrel rendelkezik, viszonylag egyszerű mechanizmusokat igényel (így könnyen implementálható), számos a módszert támogató eszköz férhető hozzá, tudásbázisát a szakértők könnyen előállíthatják és karbantarthatják. A hibafelismerő modul azonban alkalmazás-független, általános célú felépítésénél több olyan kérdés is felmerül, amely további módszerek bevonását igényli.

**Nagy méretű rendszerek**

A diagnosztika sok más, diszkrét rendszerek esetén fontos problémához hasonlóan egy, a rendszer állapotterében végzett keresést jelent. Sok komponensből álló összetett rendszerekben mind a rendszermodell kidolgozása, mind a keresés hatékony elvégzése nehéz feladat. Ebben segíthet a *Petri hálók* alkalmazása. A Petri háló modell egyesíti a szemléletes grafikus leírás és a mögöttes jól definiált matematikai formalizmus előnyeit. A modell strukturális elemei a *helyek*, *átmenetek* és a közöttük kapcsolatot teremtő irányított *élek*. A dinamikus működést *tokenek* biztosítják, amelyek az átmenetek *tüzelése* során a bemeneti helyekről a kimeneti helyekre kerülnek. A modell egy állapotát a jelölés, azaz az *n* helyen levő tokenek száma, mint egy *n* elemű vektor adja meg. A Petri háló jelöléstől nem függő tulajdonságait a *hely* és a *tüzelési invariánsok* jellemzik.

Portinale és munkatársai dolgozták ki a Petri hálók diagnosztikai felhasználásának elméletét (*Portinale,* 1993). Amennyiben a rendszer viselkedési modellje leírható a fenti, elsőrendű predikátumokat tartalmazó, negálásmentes literálokból alkotott logikai kifejezésekkel, akkor a modell egyszerű transzformációval Petri hálók alakra hozható. A *helyek* belső hibaállapotoknak felelnek meg, a meghibásodások *forrás átmenetekkel*, a megfigyelhető hibajelzések pedig *nyelő átmenetekkel* fejezhetők ki. A forrás és nyelő átmeneteket összekötő háló a hibaterjedési folyamatokat írja le. A Petri háló alapú diagnosztikai rendszermodellben a tokenek a forrásoktól indulnak, és ha a nyelőkig eljutnak (azaz a valóságban a hibák hatása eljut a felhasználóig), akkor elhagyják a hálót, ami így kiürül. A diagnosztikai probléma a transzformált modellben tehát az *átmenet invariánsok* kiszámításával oldható meg. Ehhez viszont nem szükséges a teljes állapottér bejárása, helyette mátrix eliminációs technikák használhatók. Erre a feladatra pedig számos hatékony algoritmus ismert még igen nagy mátrixméret mellett is.

**Többértékű hibamodellek kezelése**

A hagyományos színezetlen Petri hálók hátránya, hogy csak kétértékű leírást valósít meg (a tokenek színezetlenek, egy helyen vagy van token, vagy nincs). Ez már akkor is hátrányként jelentkezhet, ha a modellezett rendszer hibamodellje bináris (vagy jó, vagy rossz állapotú komponensek), hiszen sok esetben a komponensek egy részhalmazát a diagnosztika nem tudja besorolni a két kategória egyikébe sem. Ilyenkor három hibaállapotra van szükség: *jó*, *rossz* és *ismeretlen*. Az ilyen jellegű problémák kezelésére született meg a *színezett Petri háló* formalizmus, amely a tokenekben bevezeti a „színek" fogalmát. A színek értelmezése igen tág lehet, akár komplex absztrakt adattípusokat is jelölhetnek. Ennek megfelelően a tokeneken értelmezett műveletek köre és így a színezett háló modellező ereje is jelentősen kibővül. Ugyanakkor a diagnosztika szempontjából fontos invariáns számítás a színezett Petri hálókra is általánosítható, így a diagnosztikai probléma azonos módon kezelhető.

Amennyiben a rendszert alkotó komponensekre részletes, több hibamódot tartalmazó hibaszemantikai modellel jellemezhetők, akkor a problématér még kis számú

komponenst illetően is igen nagyra nőhet. Ekkor a színezett Petri hálóknál hatékonyabb modellező eszköz a *kényszerháló alapú* reprezentáció. A kényszerháló modell három fő elemből áll: az $X=\{x_1,\ldots,x_n\}$ *változók* halmaza, a változókhoz rendelt $D=\{d_1,\ldots,d_n\}$ *domének*, azaz értékkészletek halmaza, valamint a változók közti kölcsönös függést, azaz kétirányú kapcsolatot definiáló $C=\{c_1,\ldots,c_k\}$ *kényszerek* halmaza. A modell grafikus megjelenítésében a változókat csomópontok, a kényszereket pedig a csomópontok között húzott élek jelenítik meg. A *kényszerkielégítési probléma* (Constraint Satisfaction Problem, CSP) lényege a változók összes olyan lehetséges érték hozzárendelésének megkeresése, amikor a kényszerháló összes kényszere teljesül. Látható, hogy a diagnosztikai feladat egyszerűen vihető át a CSP problémakörébe: a változók jelképezik a rendszerkomponenseket, a változók értékkészlete a hibamódokat (ezek között természetesen a hibátlan működést is!), a kényszerek pedig a komponensek kommunikációs és egyéb kapcsolatait, és a hibaterjedési mechanizmusokat modellezik (*Altmann et al.,* 1996]. A megfigyeléseket speciális, egy elemű értékkészlettel rendelkező változók rögzítik. A kényszerkielégítési probléma megoldása szolgáltatja azokat a komponens hibamódokat, amelyek kompatibilisek a megfigyelésekkel, ez tehát konzisztencia alapú diagnosztikai megoldás.

**Bizonytalanság, valószínűségi folyamatok modellezése**

Szakértői rendszerekben a kezdetektől fontos kritériumként jelent meg a nem teljesen megbízható ismeretek vagy következtetések kezelése. Ilyen problémát vet fel pl. a véletlen jellegű folyamatok kezelése, igen bonyolult rendszerek közelítő leírása, vagy nem teljesen ismert jelenségek figyelembevétele. Amennyiben a bizalom mértéke az ilyen tudásbázis elemeiben (jó közelítéssel) számszerűsíthető, akkor az ismeretek valószínűségi alapú kezelését valósíthatjuk meg. Produkciós rendszerben a valószínűségi alapú következtetési folyamat kézenfekvő megvalósítása az, hogy a ténybázis elemeihez valószínűségeket, az egyes következtetésekhez pedig feltételes valószínűségeket rendelünk. Ekkor a ténybázis egyes elemeinek valószínűségét a kiinduló valószínűségekből a Bayes-tétel segítségével számíthatjuk:

$$\mathrm{P}(d_i \mid f_j) = \frac{\mathrm{P}(f_j \mid d_i)\,\mathrm{P}(d_i)}{\sum_k \mathrm{P}(f_j \mid d_k)\,\mathrm{P}(d_k)}$$

A fenti gondolatmenet általánosításaként foghatók fel a függőségeket grafikus modellben megjelenítő *Bayes hálók* (*Heckerman,* 1995). A kényszerhálókhoz hasonlóan a Bayes háló modellek is három fő elemből állnak: az $X=\{x_1,\ldots,x_n\}$ *változók* halmaza, a változókhoz rendelt $P=\{p_1,\ldots,p_n\}$ lokális *valószínűség-eloszlások*, és a változók közti feltételes függés kapcsolatot definiáló $S=\{s_1,\ldots,s_k\}$ hálózati struktúra. A modell grafikus képében a változókat csomópontok, a feltételes függés kapcsolatokat pedig a csomópontok között húzott irányított élek jelenítik meg. A Bayes-tétel segítségével a hálóban valószínűségi következtetési szabályokat alkalmazó algoritmusokat dolgoztak ki, melyek képesek a megfigyelések rögzítése után bármely változóhoz rendelt valószínűséget meghatározni.

A Bayes háló formalizmus nagy előnye a hibafa reprezentációval való szoros kapcsolata. Egyszerű algoritmussal a létező hibafák Bayes hálóvá transzformálhatók (*Portinale és Bobbio*, 1999). A transzformált modellből a Bayes hálókhoz kidolgozott algoritmusokkal ugyanazok az információk kinyerhetők, mint a hibafa modellből, de a Bayes háló általánosabb volta és nagyobb eszközkészlete révén még további információkat is szolgáltat. Ugyancsak nagy előny, hogy a Bayes hálók paraméterei tanítással a valós

mérési eredményekből automatikusan is beállíthatók és finomhangolhatók. A modell hátránya a hibafa alapú megoldással szemben kisebb elterjedtsége és a (kommerciális) megoldó eszközök hiánya.

**Időfogalom, a dinamikus viselkedés figyelembevétele**

Az idő figyelembevételével a következtetési folyamatban a *temporális logika* foglalkozik (*van Benthem,* 1995). Az idő beépítésére a logikai folyamatba két fő irányzat terjedt el: a modális logikai és a predikátum alapú megközelítés. A *modális logikai megközelítés* a természetes nyelvben előforduló idővel kapcsolatos fogalmak logikai reprezentációjával foglalkozik. Két gyenge („valamikor előfordult" és „valamikor előforduló"), továbbá két erős („mindig igaz volt, hogy", ill. „mindig igaz lesz, hogy") modális operátort definiál, s ezek logikai kapcsolatát vizsgálja. A négy modális operátorhoz később két további bináris temporális operátor („azóta, hogy" és „addig, amíg" jelentéssel) társult. A modális logikai módszert kiterjesztették *diszkrét idejű* rendszerekre is. Bár a modális megközelítésű temporális logika elemei hasznosak lehetnek a szakértői tudásbázis formalizálásában, az általa biztosított következtetési eszközök kevésbé jelentősek a diagnosztikai alkalmazásokban.

A *predikátum-elvű* temporális logikában az időfüggő jelenségeket leíró predikátumokategy további *idő attribútummal* látják el. Az időbeliséget (pl. a modális logika 4+2 operátorát) temporális predikátumok bevezetésével lehet a modellben megjeleníteni. Néhány axiómával megteremthető a temporális predikátumok kapcsolata. A megközelítés nagy előnye, hogy ezek után a következtetésre az elsőrendű predikátumkalkulus módszerei változatlan formában használhatók.

## EREDMÉNY ÉS ÉRTÉKELÉS

**A hibalokalizáló modul a Paksi Atomerőmű Reaktorvédelmi Rendszerében**

A paksi Reaktorvédelmi Rendszer (RVR) feladata a működő nukleáris reaktor blokk bizonyos kiemelt technológiai paramétereinek folyamatos megfigyelése és az egyes paraméter-konfigurációk által meghatározott vészhelyzetekben operátori riasztások kiadása, végső esetben pedig a blokk leállítása. Az RVR kialakítása a SIEMENS TELEPERM/XS (TXS) digitális folyamatirányító rendszerre épül, ami architektúra szempontjából elosztott többprocesszoros jelfeldolgozó számítógéprendszernek tekinthető. A TXS rendszerprogramja tartalmaz beépített processzor önteszt és monitorozó funkciót, valamint a felhasználói feldolgozásokat definiáló adatbázis (a TXS rendszeren „futó felhasználói applikáció") szintén megad tesztjeleket és ellenőrzési lehetőségeket (*Bokor et al.,* 1997).

Az RVR felhasználói szoftvere ezek segítségével a reaktorvédelemben bekövetkező, rendelkezésre állást befolyásoló meghibásodások által kiváltott jelzésekről és védelmi eseményekről információt gyűjt. Ezen adatok köre négy részre osztható: a rendszert alkotó számítógépek processzorainak ellenőrzése, az analóg bemeneti értékek hihetőségének vizsgálata, a digitális bemeneti értékek hihetőségének vizsgálata és a neutronfluxus (NF) jellemzők monitorozása. Az összegyűjtött információkat RVR állapot ellenőrző rendszer továbbítja az egyes kisegítő alrendszerek, többek között a karbantartási és ellenőrző feladatokat ellátó Szerviz számítógép és a rendszeradatokat megfigyelő és archiváló blokkszámítógép (BSZG) felé. A BSZG feladata a fontosabb technológiai és üzemeltetési jellemzők megjelenítése és archiválása. A BSZG-n tehát az RVR pillanatnyi állapota is megtekinthető, de a meghibásodásokról és eseményekről gyűjtött információk SQL archívumok formájában a BSZG-t alkotó szervereken

tárolódnak, továbbá (korlátozott ideig) utólag is visszakereshetők. Ezen adatok az RVR rendszer hardver és szoftver struktúrájának ismeretével kiegészítve lehetőséget adnak arra, hogy a reaktorvédelem esetleges hibáiról képet kapjunk, valós hiba bekövetkezésekor pedig a lehető legpontosabban behatároljuk annak helyét és jellegét.

**Az FMH értelmező modul és az FMH adatbázis**

Az FMH értelmező modul feladata a Feltételezett Meghibásodások Halmazát definiáló, szöveg formátumú fájlok beolvasása és feldolgozása. Az FMH adatbázisaiban definiálja az elemzésben figyelembe vett komponensek és jelek körét, továbbá megadja az egyes lehetséges hibákhoz tartozó diagnosztikai feltételrendszert. A diagnosztikai állapot és a feltételek leírására, valamint a kiértékelés mechanizmusához az ítélet, továbbá a predikátumkalkulus eszközeit használja fel. Az FMH jelenlegi formájában a következő három fő részből áll: *alapadatok*, *alapesemények*, és a *feltételezett meghibásodások*.

Az *alapadatok* jelentik azokat a diagnosztikai információkat, amelyeket a védelmi rendszer a Hibadetektáló modul számára szolgáltatni képes. Ezek elsősorban az ún. processzor státusz szavakat hordozó jelek, valamint a BSZG-t alkotó szerverek archívumaiban található további jelfelület. (A processzorok státusz szavai lokalizált hibainformációkat jelenítenek meg, így a rendszerbe épített öntesztek eredményeinek komponensekhez kötött megjelenítéseként foghatók fel.) Az *alapesemények* és a *feltételezett meghibásodások* a hibák kialakulásának feltételeit, terjedését és a kimeneten való megjelenését leíró logikai állításokból készített adatbázisok. A két adatbázis tehát felépítésében megegyezik, azonban közöttük hierarchikus viszony van: míg az alapesemény adatbázisban kizárólag alapadatok logikai kifejezései szerepelnek, addig a feltételezett meghibásodások adatbázisban szereplő kifejezések feltétel részei már az alapesemény feltételek következmény részében szereplő logikai változókból állnak. Ez a hierarchikus felépítés a nagyszámú feltételt tartalmazó logikai formula halmaz felépítését strukturáltabbá és könnyebben áttekinthetővé teszi. Ugyanakkor a tagolás a meghibásodási és hibaterjedési mechanizmusokat is jól kifejezi: az alapesemények kifejezéseinek feltétel részei a *fizikai meghibásodásokhoz* kapcsolódnak és a lehetséges hibaok hipotéziseket jelentik, míg következmény részek a meghibásodás hatására létrejövő *belső hibás rendszerállapotokat*. A feltételezett meghibásodások feltétel részei ezen hibás rendszerállapotok kombinációiból állnak és azt fejezik ki, hogyan jutnak ezek a hibák a rendszer kimenetére. A rendszerfelületen is észlelhető belső hibás rendszerállapotok eredményei a következmény részben szereplő *hibaesemények*.

Az FMH adatbázis tárolására a könnyen olvasható és karbantartható szöveges formátumot ajánljuk. Az alapadatok esetében a feldolgozott jelfelület definiálására és jellemzőinek megadására van szükség, ezért ebben az esetben egy határoló karakterrel elválasztott mezőkből álló szöveges táblázatot használhatunk. Ez a formátum egyszerű szövegszerkesztővel is létrehozható és módosítható, ugyanakkor táblázatkezelő programmal könnyen és hatékonyan manipulálható. Az alapesemények és az FMH bejegyzések esetében a produkciós rendszerekben széles körben használt „**HA** *logikai feltétel* **AKKOR** *következmény*” típusú leírás alkalmazható. A felhasznált logikai kifejezéseket a könnyen olvasható *infix* formátumban célszerű megadni. Ez a formátum a standard matematikai jelölésrendszernek felel meg, melyben a változókat operátorok választják el, a kifejezés kiértékelése balról jobbra halad, és zárójelekkel változtatható meg az operátorok kiértékelési precedenciája. Példaként bemutatjuk az FMH adatfájl egy részletét az *1. ábrán* (az első néhány sor után a fájl fennmaradó részét levágtuk és a „…” karakterrel helyettesítettük).

**1. ábra**

**A paksi Reaktorvédelmi Rendszer FMH szabálybázisának részlete**

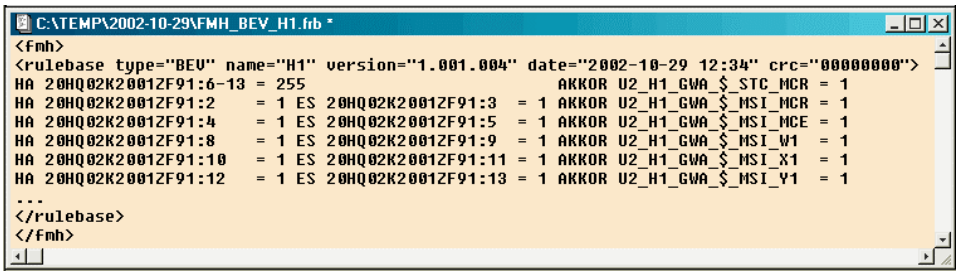


*Figure 1: Part of the FMH rule base developed for the Reactor Protection System in the Paks NPP*

Az alapadatok táblázatainak kezelése egyszerűbb, mint az alapesemények és az FMH bejegyzések adatbázisainak kezelése. Az előbbiek esetében ugyanis a táblázat minden sora egy objektumot ad meg a COSMOS objektumtárában. Így csak be kell olvasni és el kell helyezni a jelek azonosítóit, továbbá paramétereit az objektum adatbázisban, valamint ki kell alakítani a keresztkapcsolatokat az összefüggő jelek között (összetartozó jelek a három redundáns készletben, ellenőrző és forrásjelek kapcsolata stb.).

Az alapesemények és az FMH bejegyzések azonban logikai kifejezéseket tartalmaznak. Ezekben a kifejezéseket a beolvasás után értelmezni is kell, ugyanis az infix forma nem célravezető az on-line gépi feldolgozás szempontjából (számításigényes értelmezőt igényel). Ezért az infix logikai kifejezéseket az FMH értelmező átalakítja *postfix* vagy RPN (Reverse Polish Notation) alakra, mely nem tartalmaz zárójeleket, és az operandusok, továbbá operátorok a feldolgozás sorrendjében szerepelnek benne. Ennek az alaknak nagy előnye, hogy egy egyszerű veremautomatával könnyen és hatékonyan kiértékelhető.

Ezeken az adatbázisokon felül az FMH értelmező kialakít két keresztreferencia táblázatot is, melyben a kifejezések eredményéül kapott logikai változók szerepelnek. Az alapadatok jeleire és az alapesemények, továbbá FMH bejegyzések logikai változóira a konvertálás során kialakított postfix kifejezésekben ugyanis nem név szerint, hanem a képzett objektum adatbázisokban hozzájuk rendelt egyedi azonosítóval hivatkozunk.

Az alapadatok objektumait és a logikai feltételekben szereplő logikai konstansokat, továbbá változókat, valamint az objektumok kapcsolatát leíró logikai kifejezéseket, s a bennük szereplő egyéb operátorokat egységes objektum struktúrában célszerű tárolni. A flexibilis kialakítás és a könnyű hozzáférhetőség miatt célszerű az FMH objektumokat is szövegesen, XML formátumban tárolni. Ennek elterjedtsége az adattárolásban és dokumentumkezelésben egyre nagyobb, továbbá számos fejlesztőkészlet támogatja, így alkalmazása is egyszerű.

**Az FMH kiértékelő modul**

Az FMH kiértékelő modul a Hibadetektáló modul diagnosztikai „motorja”, amely az FMH alapesemények és bejegyzések segítségével, a BSZG archívumokra támaszkodva elkészíti egy kijelölt időpontban vagy időintervallumban az RVR rendszer diagnosztikai állapotképét, illetve eseménynaplóját. Ez a következő funkciók végrehajtását igényli.

Miután a felhasználó kiválasztotta a vizsgálni kívánt időtartományt, az FMH kiértékelő modul a BSZG interfész segítségével először finomítja a megvizsgálandó időintervallumot,

majd megkéri tőle az intervallumba eső események (amikor az RVR állapota megváltozott) időpontlistáját. A időpont lista minden egyes elemére elvégzi az FMH táblák kiértékelését, ehhez folyamatosan kéri le a jelek értékeit a BSZG-től. A kiértékelendő kifejezések postfix formában elrendezett objektumokból állnak, melyet veremautomatával dolgoz fel.

A postfix forma *tokenek* rendezett listája. A tokenek egyenrangúak, közöttük prioritási különbség nincs, a feldolgozási sorrendet egyedül a tokenek sorrendje határozza meg. Négyfajta token létezik: *adat tokenek*, amiket a kifejezésekben operandusaként szereplő logikai, numerikus és esetleg szöveges konstansok és változók alkotnak; *operátor tokenek*, azaz a kifejezésekben szereplő matematikai operátorok; *funkció tokenek*, a kifejezések kiértékeléséhez szükséges műveletek, továbbá *referencia tokenek*, a hivatkozásokban használt egyedi objektum azonosítók.

*Példaként bemutatjuk a paksi Reaktorvédelmi rendszer FMH szabálybázisának a H1-es kommunikációs alrendszerre vonatkozó egyik szabálya konvertálását. Az eredeti szabály az alábbi.*

| |
|---|
| … |
| HA X0YY61K400ZQ11:11-15=21 ES X0YY62K400ZQ11:11-13=7 AKKOR H1_MSI_Y1_STC_Y1=1 |
| … |

*A gépi feldolgozáshoz lefordított fájl fenti szabályhoz tartozó része magyarázattal*

| **Konvertált fájl** | **Magyarázat** |
|---|---|
| … | |
| [12;0] | Szabály fejléce: 12-es számú szabály, prioritási szint: 0 |
| 1;SIGVAL;X0YY61K400ZQ11 | 1. token: az X0YY61K400ZQ11 nevű jel értéke kerüljön a verembe |
| 2;INT;63488 | 2. token: egész konstans kerüljön a verembe, értéke 63488 (11-15 bit) |
| 3;MASK;- | 3. token: bináris maszkolási művelet a verem tetején levő két értékkel |
| 4;INT;21 | 4. token: egész konstans kerüljön a verembe, értéke 21 (kívánt eredmény) |
| 5;EQ;- | 5. token: a verem tetején levő két érték összehasonlítása, egyenlőségre |
| 6;SIGVAL;X0YY62K400ZQ11 | 6. token: az X0YY61K400ZQ11 nevű jel értéke kerüljön a verembe |
| 7;INT;14336 | 7. token: egész konstans kerüljön a verembe, értéke 63488 (11-15 bit) |
| 8;MASK;- | 8. token: bináris maszkolási művelet a verem tetején levő két értékkel |
| 9;INT;7 | 9. token: egész konstans kerüljön a verembe, értéke 21 (kívánt eredmény) |
| 10;EQ;- | 10. token: a verem tetején levő két érték összehasonlítása, egyenlőségre |
| 11;AND;- | 11. token: a verem tetején levő két Boolean érték logikai ÉS kapcsolata |
| <12;H1_MSI_Y1_STC_Y1;1> | A H1_MSI_Y1_STC_Y1 jelű alapesemény 1-es értékre állítása, ha a szabály (tokenhalmaz) kiértékelése IGAZ Boolean értékkel zárult. |
| … | |

A postfix formában adott kifejezések gépi kiértékelése egyszerűen és gyorsan elvégezhető. A veremautomata LIFO (Last In First Out) veremből és veremvezérlőből áll. A veremvezérlő beolvassa a soron következő tokent. Ha az adat token, akkor azt elhelyezi a verem tetején (a verem lefelé bővül, legfelül mindig az utoljára betett token van). Referencia tokeneket illetően hasonlóan jár el, csak itt a hivatkozási azonosítót teszi a verembe. Operátor vagy funkció token esetén a művelettől függő számú (általában egy vagy kettő) tokent vesz le a verem tetejéről. Ezután elvégzi a műveletet a leemelt értékekkel, mint operandusokkal, majd a kapott eredményt tárolja a verem tetején. Az összes token beolvasása és feldolgozása után az eredmény a verem tetején képződik.

A következőkben bemutatjuk a paksi Reprezentatív Konfigurációból származó, a tesztelés során felvett egyszerű meghibásodást és annak elemzését:

**2. ábra**

**L2-es kommunikációs kapcsolat megszakadásakor fellépő események**

| | | | | | **Érték binárisan kifejtve** | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Dátum** | **Idő** | | **Jel** | **Érték** | | AU Komm. | AU SW | | | | | | | | TsaY-TSaX | | | | | Teszt/diag. | Param eng. | AU Param |
| | | | | | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
| 2003-10-30 | 12:25:42 | 067 | Y_TSa1 | 262336 | | | | | | | | | | | | | | | | | | | | |
| 2003-10-30 | 12:25:42 | 077 | X_TSa1 | 262 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **1** | 0 | 0 | 0 | 0 | 0 | **1** | **1** | 0 |
| 2003-10-30 | 12:25:42 | 077 | Y_TSa1 | 393408 | | | | | | | | | | | | | | | | | | | | |
| 2003-10-30 | 12:25:42 | 082 | X_TSa1 | 262406 | 0 | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **1** | 0 | 0 | 0 | 0 | 0 | **1** | **1** | 0 |
| 2003-10-30 | 12:25:42 | 088 | X_TSa1 | 393478 | 0 | **1** | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **1** | 0 | 0 | 0 | 0 | 0 | **1** | **1** | 0 |
| 2003-10-30 | 12:25:42 | 098 | Y_TSa1 | 262336 | | | | | | | | | | | | | | | | | | | | |
| 2003-10-30 | 12:25:43 | 038 | X_TSa1 | 262406 | 0 | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **1** | 0 | 0 | 0 | 0 | 0 | **1** | **1** | 0 |
| 2003-10-30 | 12:26:48 | 943 | Y_TSa1 | 262208 | | | | | | | | | | | | | | | | | | | | |
| 2003-10-30 | 12:26:49 | 043 | X_TSa1 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **1** | **1** | 0 |
| 2003-10-30 | 12:26:49 | 043 | Y_TSa1 | 393280 | | | | | | | | | | | | | | | | | | | | |
| 2003-10-30 | 12:26:49 | 147 | X_TSa1 | 131078 | 0 | 0 | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **1** | **1** | 0 |
| 2003-10-30 | 12:26:49 | 297 | Y_TSa1 | 262208 | | | | | | | | | | | | | | | | | | | | |
| 2003-10-30 | 12:26:49 | 797 | X_TSa1 | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **1** | **1** | 0 |

*Figure 2: Events occurring due to the disruption of an L2-type communication connection*

A *2. ábrán* az X jelű redundáns készlet A oldali TS számítógép 1. processzora és az Y jelű redundáns készlet A oldali TS számítógép 1. processzora közti L2-es kommunikációs kapcsolat meghibásodásakor fellépő hibaesemények láthatók. A két processzor (X_TSa1-el és Y_TSa1-el jelölt) státusz szavaiban bekövetkező változások mutatnak rá a hiba okaira. Ezek közül az X készlet státusz szavát binárisan kifejtve is ábrázoltuk. Ebben a 8-as számú bit jelzi a kapcsolat megszűntét.

A diagnosztikai elemzés első lépése az archív adatok bekérése a vizsgált rendszerből a megadott időintervallumra vonatkozóan. Ezután a hibalokalizáló modul a lekérdezett adatokon végiglépkedve a szabályokban szereplő bármely jel változásakor kiértékeli a teljes szabályhalmazt és meghatározza, mely FMH bejegyzések váltak aktívvá vagy inaktívvá. Az így kapott diagnosztikai képeket (illetve azok változásait)

egyszerű, könnyen áttekinthető diagnosztikai napló formájában megjeleníti a program főablakában.

**3. ábra**

**A Hibalokalizáló modul ablaka az FMH szabálybázis kiértékelése utáni állapotban**

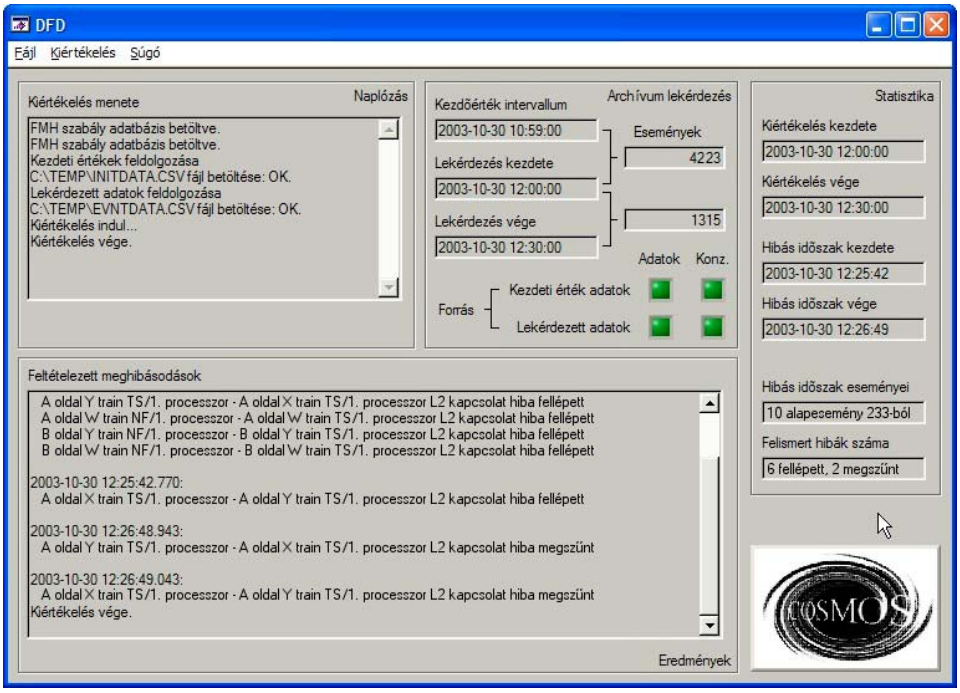


*Figure 3: The view of the Fault Localisation module after the evaluation of the FMH rule base*

A *3. ábrán* látható a hibalokalizáló modul főablaka a kiértékelés sikeres befejezése *után*. Jól megfigyelhető, hogy a teszt során előidézett meghibásodáshoz tartozó két alapesemény (az X készlet TSa1 számítógéptől induló kapcsolat #L2_TSA_X1_TSA_Y1 alapeseménye, valamint az Y készlet A oldali TSa1 számítógéptől induló kapcsolat #L2 _TSA_Y1_TSA_X1 alapeseménye) felléptét és megszűntét helyesen ismerte fel a program.

**A degradáció-elemző modul**

A modul feladata a paksi Reaktorvédelmi Rendszer (RVR) működésbiztonsági mutatóinak meghatározása. Ennek alapja a védelmi rendszerre felépített, hibafák formájában megadott megbízhatósági modell analízisén alapul. A hibafa analízis (*Lee et al.,* 1985) képes meghatározni, hogy a rendszer egyes komponensének meghibásodása után (hibafa alapesemény), hogyan változik meg egy magas szintű rendszerjellemző (csúcsesemény), amelyre a hibafát építették. Ebben az esetben az egyik ilyen csúcsesemény az *RVR beavatkozása elmaradásának* valószínűsége. Ez azt jelenti, hogy ha a reaktorblokk olyan állapotba kerül, hogy be kell avatkozni a működésébe (pl. le kell

állítani), akkor az RVR meghibásodásakor hogyan változik a valószínűsége annak, hogy ez a beavatkozás elmarad. A degradáció-elemző modulnak ismernie kell az RVR-ben fellépett hibát, amelyet az előzőekben ismertetett „Hibalokalizáló" modultól kap meg. A hiba ismeretében az egyes csúcseseményekre előre megszerkesztett hibafák alapeseményeit módosítja és újraszámítja a csúcsesemények bekövetkezési valószínűségeit.

**A működés koncepciója: off-line műveletek**

A degradáció-elemző modul számos olyan ismeretet használ fel, amelynek összegyűjtése, megadása nem a kiértékelés időpontjában, hanem azt megelőzően, az ún. *off-line* fázisban történik. Ezzel szemben azokat a műveleteket, amelyeket már egy konkrét hibaszituáció megbízhatósági kiértékelése során kell elvégeznünk, on-line műveleteknek hívjuk. A degradáció-elemző modul off-line fázisában elvégzendő tevékenységek a következők.

- A Feltételezett Meghibásodások Halmazának (FMH) definiálása. Az FMH azokat az egyedi meghibásodásokat tartalmazza, melyeket a hibadetektáló modul azonosítani képes.
- Az Alapesemény Halmaz (AH) definiálása. Az AH tartalmazza mindazokat az egyedi komponens meghibásodásokat vagy komponens meghibásodási módokat, amelyekről közvetlenül valószínűségi információkkal (modell, paraméterek /meghibásodási ráta, javítási idő, tesztelési intervallum stb./, esetleg paraméterbizonytalanság) rendelkezünk. Optimális esetben AH megegyezik FMH-val, de ez nem szükségszerű.
- A keresendő csúcsesemények definiálása és valószínűségi limitek létrehozása. Az analízis által szolgáltatandó végeredményekhez tartozó magas szintű események kiválasztása általában más típusú analízisek, pl. FMEA segítségét igénylik. Fontos lépés az elfogadási küszöbértékek (limitek) megadása, mert ezekhez viszonyítva lehet majd az egyes meghibásodások hatásait minősíteni.
- A hibafák felépítése és tárolása az AH elemei segítségével a kiválasztott csúcseseményekből kiindulva, a RiskSpectrum (Relcon AB) hibafa szerkesztő moduljának felhasználásával. Fontos szempont, hogy az FMH elemei vagy alapesemény szinten, vagy hibafa-kapu szintjén beépítésre kerüljenek a hibafákba. Hibacsoportok létrehozása az FMH elemeiből a hibafigyelmeztető funkció aktiválásához.
- A hibafák és az FMH konzisztenciájának ellenőrzése (ellenőrző modul). Az FMH elemek azonosítóinak meg kell egyezniük a megfelelő hibafa-alapesemény vagy kapu azonosítókkal. A hibafákban minden FMH elemhez hozzá kell rendelni legalább egy alapeseményt vagy kaput.
- A minimális vágatok generálása a felépített hibafákra (előfeldolgozás). Ezeket a műveleteket a degradáció elemző modul hajtja végre a RiskSpectrum hibafa analízis moduljának meghívásával. Mivel a RiskSpectrum hibafa analízis modulja a csúcsesemény bekövetkezési valószínűségeket két lépésben számítja (MCS, majd időfüggő analízis), az on-line feldolgozás jelentősen gyorsítható az első lépés elő-feldolgozásként való elvégzésével.
- Egyszeres meghibásodásokhoz tartozó megbízhatósági értékek legenerálása a RiskSpectrum hibafa-analízis moduljának felhasználásával. Ugyancsak elő-feldolgozásként létrehozható az egyedi meghibásodások hatását tartalmazó táblázat (1000 alapesemény esetén 1000 sor).

Ha egy egyedi meghibásodás lép fel a rendszerben, nem szükséges a hibafák on-line kiértékelése, elég az előre legenerált táblázat megfelelő sorának kikeresése. A többszörös hibák (hibakombinációk) hatása azonban már sajnos nem számítható, így ilyenkor létjogosultsága lehet az on-line kiértékelésnek, bár a feldolgozás ilyenkor is gyorsítható ismételt (kvázi on-line) elő-feldolgozással. A legtöbb esetben a többszörös hibák nem egyszerre lépnek fel, hanem először az első meghibásodás jelentkezik, majd egy bizonyos idő elteltével ehhez adódik a következő hiba hatása. Az első meghibásodás észlelésekor nem szükséges az on-line kiértékelés, csak az előre generált táblázatból keressük ki a valószínűségi értéket. E mellett azonban már fel lehet készülni a következő (esetleg be sem következő) hibára is: a már fennálló meghibásodás mellé az összes többi lehetséget véve ismét generálható egy, az előbbihez hasonló táblázat, amely rendre két meghibásodás hatását tartalmazza. A feldolgozásnak ezekben az esetekben eseményvezérelteknek kell lenniük, ciklikus hibafa-kiértékelésre, valamint felhasználó által indított kiértékelésre nincs szükség (opcióként megvalósítható, de felesleges). Az elő-feldolgozást a degradáció-elemző modul irányítja.

**4. ábra**

**Az aktuális erőművi állapot a RiskSpectrum hibafa modell nézetében**

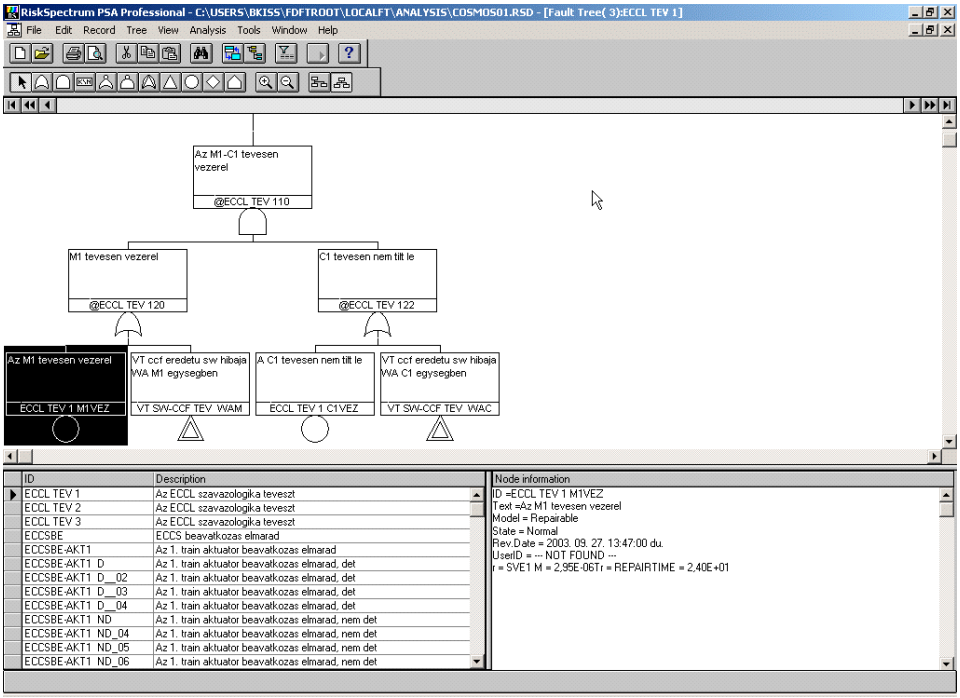


*Figure 4: RiskSpectrum fault tree model view of the actual plant status*

**On-line műveletek**

Amennyiben úgy következik be többes meghibásodás, hogy a szükséges táblázatok még nem állnak rendelkezésre, a hibafa on-line kiértékelése szükséges a RiskSpectrum hibafa

analízis moduljának on-line meghívásával. A kiértékelés várható ideje nagy rendszerekben néhány percet is igénybe vehet, ilyenkor az analízis futását, illetve a korábbi valószínűségi érték érvénytelenségét jelezni szükséges. Az on-line kiértékelés alapja a hibafa-adatbázis módosítása. Amennyiben a bekövetkezett meghibásodás (FMH elem) megfeleltethető egy alapeseménynek a hibafában, akkor az adott alapesemény valószínűségi modelljét (függetlenül annak korábbi voltától) konstans valószínűségi modellre kell cserélni, és konstans valószínűségi paraméterként 1-es értékkel kell ellátni (biztosan bekövetkezett esemény). Amennyiben az FMH elem hibafa kapunak felel meg (ez egyszerű kereséssel megállapítható az adatbázisban), az összes adott kapuhoz tartozó alapeseményen el kell végezni az átállítást (valójában a hibafán visszafelé „le kell menni" az alapeseményekig). Ennek a folyamatnak egy állapota figyelhető meg a *4. ábrán*, ahol egy, a Hibalokalizáló modul által felderített hibának megfelelő hibafa alapesemény beállítása történt meg.

Ennek elvégzése után indítható a hibafa analízis modul. A hibafák módosításáról és az analízis indításáról a degradáció elemző modul automatizált módon gondoskodik az állapotkép alapján.

**Ciklikus (idővezérelt) műveletek**

Megadott időegységenként a hibafigyelmeztető modul meghívása. A modul feladata az FMH elemek bekövetkezésekor az időpont tárolása, valamint minden egyes bekövetkezéskor határérték-figyelési számítás indítása. A repülésben már régóta alkalmazott határérték-figyelés lényege az, hogy a már bekövetkezett meghibásodások trendjeit felállítva megállapítható, mikor nő meg kiugróan (és rendszerszintű beavatkozást igénylően) a meghibásodásszám. Ciklikusan elvégzendő feladat a határérték túllépés számítása (határérték-túllépés: adott időegység alatt a meghibásodások száma meghaladja a korábbi statisztikai adatok alapján képzett határértéket), valamint a határértékek dinamikus újraszámítása.

## KÖVETKEZTETÉSEK

A bonyolult, nagy terjedelmű, biztonságkritikus berendezések meghibásodása elkerülhetetlen. A bekövetkező hiba megváltoztatja a rendszer képességeit, még akkor is, ha az hibatűrő. A megváltozott helyzetben elengedhetetlen, hogy minél gyorsabban lokalizálni lehessen a hibát, majd ezen információ birtokában el kell dönteni, hogy hogyan változtak meg a rendszer képességei. Az általunk kidolgozott eljárás a Paksi Atomerőmű Reaktorvédelmi Rendszer rendszerének meghibásodásakor képes lokalizálni a hibát, majd hibafa analízissel kiszámítja a rendszer megváltozott képességeit, elsősorban a védelmiek degradációját. A fejlesztés eredményeként létrejött „Hibalokalizáló és degradáció-elemző" rendszer használatával a blokki operátori személyzet képes eldönteni, hogy milyen feltételekkel üzemeltetheti tovább a reaktorblokkot, ha a reaktorvédelmi rendszer meghibásodott.

## KÖSZÖNETNYILVÁNÍTÁS

A cikkben bemutatott kutatási és fejlesztési eredmények a COSMOS projekt keretében készültek, amelyet a Magyar Oktatási Minisztérium támogatott a Nemzeti Kutatási és Fejlesztési Program keretében (NKFP) 2/016/2001.

## IRODALOM

Altmann, J. Pataricza, A. Bartha, T. Urbán, P. (1996). "Constraint Based System-Level Diagnosis of Multiprocessors", 2nd European Dependable Computing Conference {EDCC-2}, 1150, Springer-Verlag Inc., A. Hlawiczka, J.G.S. Silva, L. Simoncini, (Eds.), 403-425.

van Benthem, J., "Temporal Logic", D.M. Gabbay, C.J. Hogger, J.A. Robinson (Eds.) (1995). Handbook of Logic in Artificial Intelligence and Logic Programming, Oxford: Clarendon Press, 241-350.

Bokor J., Szabó G., Gáspár P., Hetthésy J. (1997). Reliability Analysis of Protection Systems in NPPs Using Fault-Tree Analysis Method. In: Proceedings of the IAEA Symposium on Computerized Reactor Protection and Safety Related Systems in Nuclear Power Plants, Budapest, 91-104.

Fensel, D., Benjamins, R. (1996). "Assumptions in model-based diagnosis", B.R. Gaines, B.R. Musen, M.A. (Eds.) (1996). Proceedings of the 10th Banff Knowledge Acquisition for Knowledge-based Systems workshop, KAW'96, 5. 1-18. Calgary: SRDG Publications, Department of Computer Science, University of Calgary.

Heckerman, D. (1995). "A Tutorial on Learning Bayesian Networks", Technical Report, no. MSR-TR-95-06, Microsoft Research, Advanced Technology Division, March.

Lee, D.W., Gros, L., Tillman, F.A, Lie, C.H. (1985). Fault Tree Analysis, Methods, and Applications – a Review. IEEE Trans. on Reliability 34. 194-203.

Peter, J., Lucas, F. (1998). "Analysis of Notions of Diagnosis", Artificial Intelligence, 1-2. 295-343.

Poole D.L. (1988a). „A Logical Framework for Default Reasoning", Artificial Intelligence, 1. 27-47.

Poole D.L. (1988b). „Representing Knowledge for Logic-Based Diagnoses", Proc. International Conference on Fifth Generation Computing Systems, 1282-1290.

Portinale, L. (1993). "Exploiting T-invariant Analysis in Diagnostic Reasoning on a Petri Net Model", Application and Theory of Petri Nets, 339-356.

Portinale, L., Bobbio, A. (1999). "Bayesian Networks for Dependability Analysis: an Application to Digital Control Reliability", Proceedings of the 15th Int. Conference on Uncertainty in Artificial Intelligence (UAI-99), 551-558.

Levelezési cím (*corresponding author*):

**Varga István**
Magyar Tudományos Akadémia
Számítástechnikai és Automatizálási Kutató Intézete
Rendszer és Irányításelméleti Kutató Laboratórium
1111 Budapest, Kende u. 13-17.
*Systems and Control Laboratory*
*Computer and Automation Research Institute*
*Hungarian Academy of Sciences*
*H-1518 Budapest, Pf. 63.*
Tel.: +36-1-279 6227, fax.: +36-1-466 7503
E-mail: ivarga@sztaki.hu