



Diagnosztikai feladatok megvalósítása ágens-alapú technikával

Németh E., Lakner¹ R., Hangos K.M.

Magyar Tudományos Akadémia Számítástechnikai és Automatizálási Kutató Intézete,
Rendszer- és Irányításméleti Kutató Laboratórium, Budapest, 1111 Kende u. 13-17.

¹Pannon Egyetem, Számítástudomány Alkalmazása Tanszék, Veszprém, 8200 Egyetem u. 10.

ÖSSZEFOGLALÁS

Dinamikus szimulátorral és adatbázis szolgáltatásokkal összekapcsolt Protégé-JADE-JESS környezetben megvalósított multi-ágens alapú diagnosztikai rendszert mutat be a cikk. Ez az implementációs környezet lehetővé teszi a heterogén információforrásokat felhasználó diagnosztikai módszerek kombinációjának használatát. A diagnosztikai rendszer általános és moduláris alkalmazhatóságát elősegítve ontológiákat definiálunk. Egy granulátor üzem meghibásodásainak diagnosztizálásán keresztül a cikk ismerteti a diagnosztikai rendszer működését, amely a HAZOP és FMEA analíziseken alapszik. (Kulcsszavak: multi-ágens rendszer, diagnosztika)

ABSTRACT

Diagnostic task realization by agent-based technics

E. Németh, R. ¹Lakner, K.M. Hangos

Systems and Control Laboratory Computer and Automation Research Institute, Hungarian Academy of Sciences
Budapest, H-1111 Kende u. 13-17.

¹Department of Computer Sciences, University of Pannonia, Veszprém, H-8200 Egyetem u. 10.

A multi-agent diagnostic system implemented in a Protégé-JADE-JESS environment interfaced with a dynamic simulator and database services is described in this paper. The proposed system architecture enables the use of a combination of diagnostic methods from heterogeneous knowledge sources. In order to facilitate the modularity and general applicability of the diagnostic system, ontologies are defined. The diagnostic system is demonstrated on a case study for diagnosis of faults in a granulation circuit based on HAZOP and FMEA analysis.

(Keywords: multi-agent system, diagnosis)

BEVEZETÉS

A bonyolult és nagy méretű biztonságkritikus üzemek berendezéseinek meghibásodása elkerülhetetlen. A bekövetkező hiba megváltoztatja a rendszer tulajdonságait, ezáltal a működését. Hibás működés esetén elengedhetetlen, hogy minél gyorsabban és pontosabban lokalizálni lehessen az esetleges hibát, majd ezen információ birtokában el kell dönteni, hogy ez milyen káros hatással lehet a jövőre nézve, illetve az időben felismert hibát lehet-e (illetve hogyan lehet) korrigálni.

Mint legtöbb mérnöki területen, így a folyamatrendszerekben is fontos szerepet tölt be és az utóbbi időben egyre inkább előtérbe kerül a hibadetektálás és -diagnosztika. Az

abnormális (rendkívüli) események kezelésének (Cameron and Raman, 2005) manapság szintén nagy figyelmet szentelnek. A rendkívüli események kezelése alatt egy rendszerben az előforduló hibák abnormális feltételeinek időben történő detektálását, diagnosztikáját és korrigálását értjük.

Bonyolult folyamatrendszerek esetén a modell alapú analitikus és heurisztikus technikák kombinációját használó diagnosztikai rendszer szükséges. Egy ilyen összetett, különböző ismeretforrásokból származó információkat felhasználó diagnosztikai feladat megvalósításának ígéretes módszere a mesterséges intelligencia területén alkalmazott multi-ágens rendszereket (Jennings and Wooldridge, 1998) felhasználó megközelítés. Multi-ágens rendszer alapú megközelítéssel leírhatóak és együtt kezelhetőek a rendszermodellek, a megfigyelések, valamint a diagnosztikai és a veszteségmegelőző eljárások. Ezért célunk egy olyan multi-ágens architektúra kidolgozása volt, amely a dinamikus modellek és a heurisztikus működési ismeretek együttes figyelembe vételével valósítja meg folyamatrendszerek hibadiagnosztikai feladatait.

A HIBADETEKTÁLÁS ÉS –DIAGNOSZTIKA FELADATAI

A hibadetektáló és -diagnosztikai módszerek (Blanke et al., 2003) három fő csoportba sorolhatók: a modell nélküli, a modell alapú és a tudásalapú módszerek. Amíg modell nélküli módszerek nem használják a rendszer modelljét, a modell alapú diagnosztika a folyamatok és meghibásodások matematikai modelljeinek felhasználásával a jel- és folyamatanalízisen alapuló analitikus redundanciát (Patton et al., 1989) vizsgálja. A tudásalapú diagnosztikai módszerek legfontosabb elemei a folyamatrendszerről rendelkezésre álló heurisztikus ismeretek és a megfigyelt szimptómák (azaz a folyamat jellegzetes mérhető változóinak a normális működésbeli referencia értékétől való eltérései).

Hibadetektálás, -diagnosztika és veszteségmegelőzés

A hibadetektálásban és -diagnosztikában alkalmazott tudásalapú módszerek a rendszer minden meghibásodási módjához egy vagy több ún. *gyökér okot* rendelnek. Ezen gyökér okok bekövetkezésének egy variációja adja egy hiba okát. A gyökér okok gyakran nem mérhetőek és diszkrét értékűek (indikátor változók), így egy gyökér ok rendszerelméleti szempontból úgy írható le, mint egy nem mérhető zavarás egy diagnosztikai célú folyamatrendszerben.

Egy mérhető vagy számítható mennyiségeken definiált relációt *szimptómának* nevezünk, ha kapcsolódik egy tetszőleges meghibásodás vagy hiba egy gyökér okához. A szimptómák működési szempontból felismert devianciák, amelyeket a rendszer dinamikus viselkedése következtében időfüggő módon azonosíthatunk. Egy szimptóma definíciójában szereplő relációk leggyakrabban egyenlőtlenségekként jelennek meg. A szimptómák értékkészlete a logikai értékek halmaza (igaz vagy hamis). Egy egyszerű szimptóma például a $temperature_{high} = (T > 1000 \text{ K})$, amelyet a mérhető T hőmérséklet segítségével definiálunk. Dinamikus rendszerek esetében a mérhető mennyiségek többsége időben változó értéket vesz fel, ezért egy szimptóma értéke (vagy jelenléte) szintén változhat az idő függvényében.

A rendszer viselkedését minden figyelembe vett meghibásodási módjában leíró dinamikus rendszermodell segítségével *predikcióval* meghatározható(ak) egy hiba vagy *meghibásodás (időbeli) következménye(i)*. A predikció megvalósítható egy olyan szimulációval, amely megjósolja a meghibásodott rendszer viselkedését. A súlyos

és/vagy kockázatos következmények elkerülésére ajánlott *megelőző beavatkozások* szintén tervezhetők és/vagy tesztelhetők szimulációval.

Gyakran nem elég, hogy felismerjük és izoláljuk egy rendszer hibás állapotát, hanem arra is szükség lehet, hogy tanácsokat adjunk a működtető személynek, hogy megfelelő *megelőző beavatkozás(ok)* kiválasztásával hogyan kerülje el a hiba nem szándékolt következményeit. Minden egyes (gyökér okával azonosítható) hibához rendelhető egy vagy több kitüntetett bemeneti jel, amely a rendszert a normális működési tartomány irányába mozdítja el, s ezzel lehetőséget teremt a súlyos következmények megelőzésére. Ezekben az esetekben további „what if” (mi lenne akkor, ha) típusú feltételes predikciók szükségesek egy veszteségmegelőző beavatkozás hatásának vizsgálatához.

Veszélyelemzés, veszélyazonosítás: HAZOP és FMEA analízis

A hibadetektálási és -diagnosztikai feladatokhoz (*Ungar and Venkatasubramanian, 1990*) szükséges információk eltérő karakterisztikával jellemezhető különféle forrásokból nyerhetők ki. Ezek az információforrások tartalmazzák a koncepcionális tervezési tanulmányokat és a kockázatelemzést, a részrendszerek vagy konkrét működési módok részletes dinamikus modelljeit, továbbá operátoroktól és egyéb üzemi dolgozóktól származó heurisztikus működtetési tapasztalatokat. A heurisztikus információk beszerezhetők a veszélyek azonosítása és elemzése, valamint a károk felmérése és csökkentése során, felhasználva az úgynevezett folyamat működőképességi elemzés (Process Hazard Analysis, PHA) módszerét. Többféle módszert használnak a PHA tanulmányokban, úgy mint a működőképesség és veszélyelemzés (HAZard and Operability Analysis, HAZOP), hibafa-elemzést (Fault-Tree Analysis, FTA), eseményfa-elemzést (Event-Tree Analysis, ETA), meghibásodásmód és -hatás elemzést (Fault Mode Effect Analysis, FMEA).

A gyakorlatban igen elterjedt a veszélyelemzési módszerek között a *működőképesség és veszélyelemzés* vagy más néven működésbiztonsági veszélyelemzés (HAZard and Operability analysis, HAZOP) (*Crawley and Tyler, 2000*). A HAZOP elemzés során több műszaki tudományterület képviselőiből álló munkacsoport kreatív és módszeres megközelítést alkalmaz azoknak a veszélyeknek és üzemeltetési problémáknak a feltárásához, amelyek a rendeltetészerű, normális működéstől való eltérésből erednek, és amelyek káros következményekkel járhatnak. A HAZOP elemzésnek az az elve, hogy a rendszer paramétereinek vagy változóinak normális állapottól való eltérését a már létező vagy kialakulóban lévő hibák okozzák. Az elemzés során előre meghatározott, ún. vezérszavakat (guide words) (pl. *MORE, LESS, NONE, ...*) használnak. Ezeket a vezérszavakat az üzem folyamatábrája szerinti különböző területeken alkalmazzák és meghatározott folyamatjellemzőkkel kombinálva állításokat fogalmaznak meg a rendeltetészerű üzemi működéstől való eltérés meghatározása érdekében. A HAZOP elemzés során felsorolják a potenciális hiba okokat és a következményeket, valamint a hibákhoz rendelhető megelőző/védelmi intézkedéseket az általános tapasztalatok alapján. A HAZOP elemzés eredményét rendszerint táblázatos formában foglalják össze. Egy példa látható az esettanulmányban szereplő *5. ábrán*. A módszer meglehetősen időigényes és ebből következően igen költséges. A vizsgálat rendszerszintű (és nem rendszerelem szintű), és ennél fogva alapvetően magára a technológiára (és nem pl. a gépészetre) irányul.

A *meghibásodásmód és -hatás elemzése* vagy más néven hibamód és hatáselemzés (Fault Mode Effect Analysis, FMEA) (*Federal Aviation Administration, 2000*) tetszőleges rendszerek, alrendszerek, berendezések, funkciók, technológiai eljárások

diagnosztikai szempontú minőségi analízise. Elsősorban mechanikai és villamos berendezések meghibásodásának vizsgálatára használják, ellentétben a HAZOP módszerrel, amely a rendszerben zajló folyamatok egymásutánosságát, ok-okozati kapcsolatait elemzi. Az FMEA feltérképezi maguknak a berendezéseknek, alrendszereknek a lehetséges meghibásodását, és a meghibásodások helyi és rendszer szintű következményeit. Az egyes meghibásodásokat a rendszeren belüli többi meghibásodástól független eseménynek tekinti, kivéve azokat a hatásokat, amelyeket maga a meghibásodás okozhat. Az FMEA analízis eredményét táblázatos formában rögzítik az 5. ábrán látható struktúrában.

DIAGNOSZTIKAI FELADATOK ÁGENS-ALAPÚ MEGVALÓSÍTÁSA

Az alkalmazott eszközök bemutatása

Az irodalomból ismert számos ágensépítő és -szimuláló szoftver (pl. ABLE, AgentBuilder, FIPA-OS, JADE, ZEUS) közül a JADE (Java Agent DEvelopment Framework) (JADE, 2005) keretrendszert választottuk multi-ágens implementációs eszközként. Ennek a nyílt forráskódú Java alapú multi-ágens fejlesztő csomagnak az előnye, hogy támogatja a Foundation for Intelligent Physical Agent (FIPA) ágens szabványát és integrálható a Protégé (Protégé, 2004) ontológia szerkesztővel és a Java Expert System Shell (JESS) (JESS, 2005) következtető rendszerrel. Az ágensek közötti kommunikáció FIPA Agent Communication Language (FIPA ACL) formában reprezentált üzenetek küldésével valósul meg.

A JADE nem tartalmaz következtetési technikákat, de integrálható több következtető rendszerrel, például a JESS-szel és a Prolog-gal is. A JESS egy következtető gép és szkript környezet, amely JAVA nyelven íródott. Rendelkezik egy nagyon hatékony Rete algoritmust használó előrehaladó és egy visszafelé haladó következtető mechanizmussal is. A diagnosztikai rendszer általános és moduláris alkalmazhatóságát biztosító ontológiák formájában leírt ismereteket használjuk fel a JADE és JESS által használt objektum példányok illetve tények előállítására. A Protégé ontológia szerkesztő eszközzel megvalósított ontológia az ún. Ontology Bean Generator segédprogram segítségével automatikusan generálható a JADE illetve JESS által kezelhető formában.

A szimulációkhoz használt dinamikus modelleket MATLAB-ban valósítottuk meg, így a MATLAB szolgáltatja a szimulált folyamatrendszer valós idejű adatait, és tartalmazza a predikcióhoz használt egyszerűsített modelleket is. A MATLAB és a JADE közötti kommunikációt a szabványos TCP/IP protokoll felhasználásával valósítottuk meg. A nagy mennyiségű adat archiválására MySQL adatbázisokat használtunk. A JADE és a MySQL adatbázisok közötti kapcsolatot a MySQL Connector/J alkalmazásával realizáltuk.

A JADE környezetben megvalósított multi-ágens diagnosztikai rendszer fő elemei és a szoftver struktúrája az 1. ábrán látható.

A diagnosztikai rendszer tudásreprezentációja

Annak érdekében, hogy megkönnyítsük a rendszer általános és moduláris alkalmazhatóságát, két ontológia halmazt definiálunk (Lakner et al., 2006a, 2006b):

1. A *folyamat-specifikus ontológia*, amely leírja a folyamatrendszerek alapvető fogalmait, azok szemantikus kapcsolatait és megszorításait, amely hasonlít az OntoCAPE projekt (Yang et al., 2003) során kifejlesztett, a folyamatrendszerek általános leírására javasolt ontológiára. A folyamat-specifikus ontológiának két része

van: a folyamatrendszerek általános viselkedésére vonatkozó közös ismeretek és az alkalmazás-specifikus tudás. Ez a leírás határozza meg a szóban forgó rendszert leíró folyamatmodell szerkezetét, és lehetőséget teremt egy olyan konkrét folyamatmodell realizáció megvalósítására, amely felhasználható valós idejű szimulációhoz és/vagy predikció alapú diagnosztikához.

2. A *diagnosztikai ontológia*, amely leírja a diagnosztikai fogalmak (pl. szimptómák, okok, gyökér okok), valamint a különböző diagnosztikai eszközök (úgy mint FMEA és HAZOP táblák) és eljárások (mint például következtetés FMEA és/vagy HAZOP tudáson) szemantikus ismereteit. Az alkalmazás-specifikus tudás tartalmazza a konkrét rendszer viselkedéséről rendelkezésre álló – az emberi szakértelem és működtetési ismeretek felhasználásával összegyűjtött – lehetséges meghibásodásokat a hozzá tartozó okokkal, következményekkel és lehetséges korrekciókkal együtt. A 2. ábrán látható a Protégé ontológia editor segítségével készített diagnosztikai ontológia egy részlete.

1. ábra

A multi-ágens diagnosztikai rendszer struktúrája

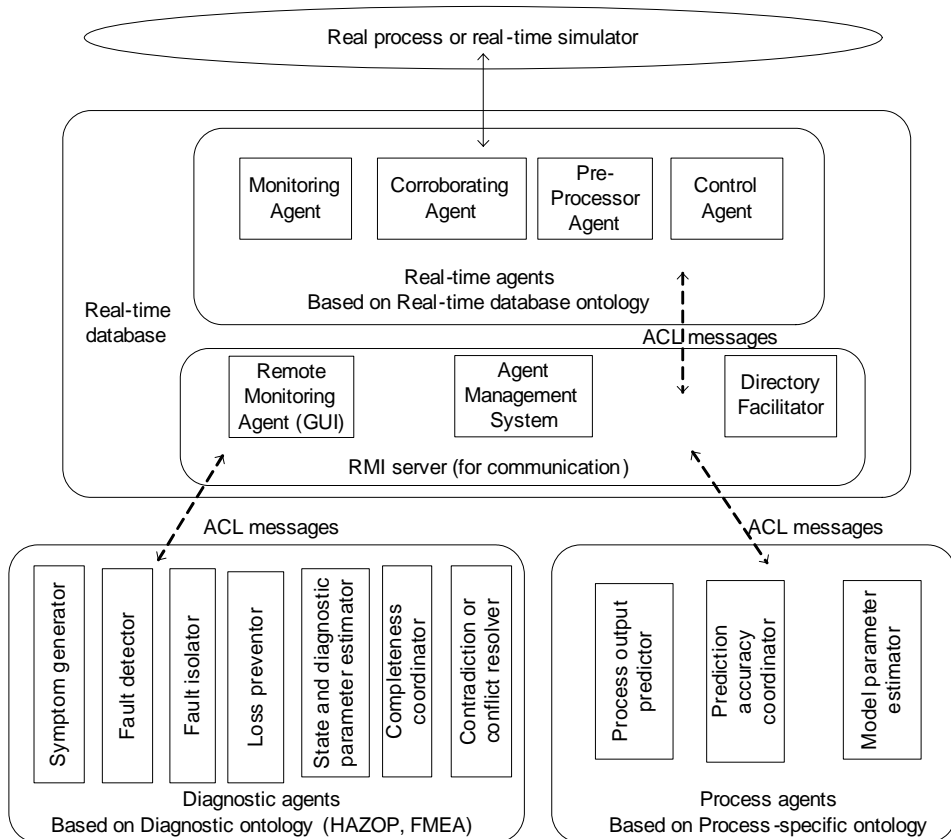


Figure 1: The structure of the multi-agent diagnostic system

2. ábra

A diagnosztikai ontológia

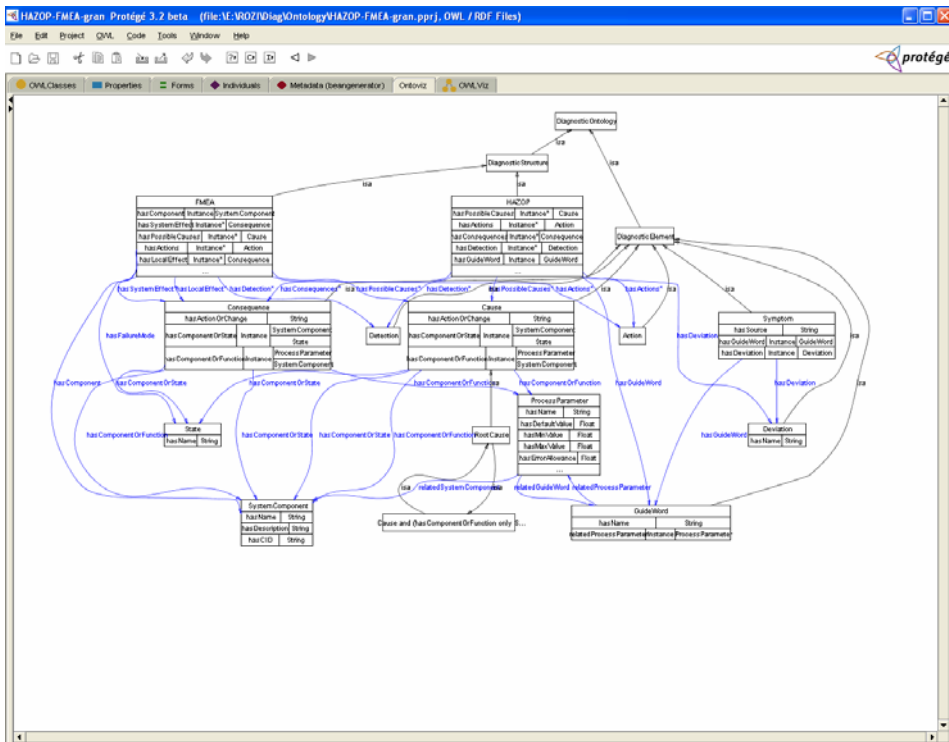


Figure 2: Diagnostic ontology

Valós idejű adatbázis Mindkét ontológia tartalmaz időben változó elemeket is, mint például a folyamatváltozók, a beavatkozó változók és a hozzájuk kapcsolódó változók. Ezen változók – egy valós rendszer vagy egy szimulátor által szolgáltatott – értékei egy *valós idejű adatbázisban* kerülnek tárolásra.

A multi-ágens diagnosztikai rendszer struktúrája

Hasonlóan a fentebb leírt ontológiák osztályozásához, a diagnosztikai rendszerhez tartozó ágensek is három fő csoportba sorolhatók (Lakner et al., 2006a,2006b):

1. A folyamatrendszer dinamikus szimulációját *folyamat ágensek* segítségével valósítjuk meg, amelyek különböző hibás és nem hibás üzemm állapotokat leíró körülmények között működnek. Néhány tipikus folyamat ágens a következő:
 - *Folyamat kimenet előrejelzők* (Process output predictors, PPs) szolgáltatják a dinamikus szimulációval előállított predikciót megelőző beavatkozásokkal vagy azok nélkül.
 - *Predikció-pontosság koordinátor* (Prediction accuracy coordinator, PAC) ellenőrzi az előrejelzés eredményének pontosságát és ha szükséges, akkor további ágenseket hív meg az eredmény finomítása érdekében.

- *Modell paraméter becslők* (Model parameter estimators, MPEs) társulnak minden egyes Folyamat kimenet előrejelzőhöz (PP). A Predikció-pontosság koordinátor felkérheti ezt az ágenszt modell paraméterek finomítására, ha az ágens előrejelzésének pontossága nem kielégítő.
2. A diagnosztikához kapcsolódó *diagnosztikai ágensek* a mérések kezdeményezését, a szimptóma felismerést, a hibadetektálást, a hiba izolálását és a nem szándékolt következmények elkerülése végett tanácsok előállítását végzik. Ezek az ágensek logikai következtetéseket és/vagy numerikus számításokat végeznek. Az alábbiakban kiemelünk néhány diagnosztikai ágens típust a diagnosztikai keretrendszerből a működésükre vonatkozó rövid feladat leírással:
- *Szimptóma generátor és státusz kiértékelő* (Symptom generator and status evaluator) működése a nem megengedett eltéréseken alapszik, feladata egy szimptóma jelenlétének ellenőrzése.
 - *Állapot és diagnosztikai paraméter becslők* (State and diagnostic parameter estimators, SPEs) segítik a szimptóma generátorokat, amelyek néhány kapcsolódó jelet és egy részrendszer egy dinamikus állapottér modelljét használják fel egy összetett szimptóma létrehozásához.
 - *Hiba detektorok* (Fault detectors, FDs) az Állapot és diagnosztikai paraméter becslők (SPEs) vagy a Modell paraméter becslők (MPEs) szolgáltatásait használják fel fejlett jelfeldolgozó módszereket alkalmazva hibák detektálásához.
 - *Hiba izolátorok* (Fault isolators, FIs) egy szimptóma bekövetkezésekor működnek annak érdekében, hogy a hibát izolálják különböző technikák (hibafa, HAZOP, FMEA, hibaérzékenység megfigyelők, stb.) segítségével.
 - *Veszteségmegelőzők* (Loss preventors, LPs) megelőző beavatkozás(oka)t illetve javító beavatkozás(oka)t ajánlanak különböző Hazard Identification Analysis (HAZID) módszerek (pl. HAZOP, predikció) segítségével.
 - *Teljesség koordinátor* (Completeness coordinator) ellenőrzi az eredmény (detektálás, izoláció vagy veszteség megelőzés) teljességét és amennyiben szükséges, további ágenseket aktivál.
 - *Ellentmondás vagy konfliktus feloldó* (Contradiction or conflict resolver, CRES) ellentmondást tartalmazó információk detektálása esetén az ellentmondást további adatok kérésével és felhasználásával, további módszerek vizsgálatával, illetve az operátor javaslatának figyelembe vételével kezeli.
3. A folyamat irányítását és monitorozását megvalósító *valós idejű kiszolgálásokhoz kapcsolódó ágensek* a következők:
- *Monitorozó ágensek* (Monitoring agents) a való világból vagy szimulációból kérnek és/vagy szolgáltatnak adatokat.
 - *Előfeldolgozó ágensek* (Pre-processor agents) felismernek nem megengedhető viselkedéseket, amelyek szimptómák lehetnek.
 - *Beavatkozó ágensek* (Control agents) a veszteségmegelőző (LP) ágensek által javasolt megelőző beavatkozásokat végrehajtják.
 - *Megerősítő ágensek* (Corroborating agents) a diagnosztikai ágensektől fogadnak el kéréseket és további mért értékeket vagy információkat szolgáltatnak.

ESETTANULMÁNY

Az ismertetett módszert és a prototípus multi-ágens diagnosztikai rendszert egy tipikus granulátor körön – amelynek főbb elemei a granulátor dob, szárító, rosta illetve törő –

mutatjuk be (Cameron et al., 2005). A granulátor kör technológiai folyamatábrája a 3. ábrán látható.

3. ábra

A diagnosztikai következtetés lépései a granulátor kör folyamatábráján

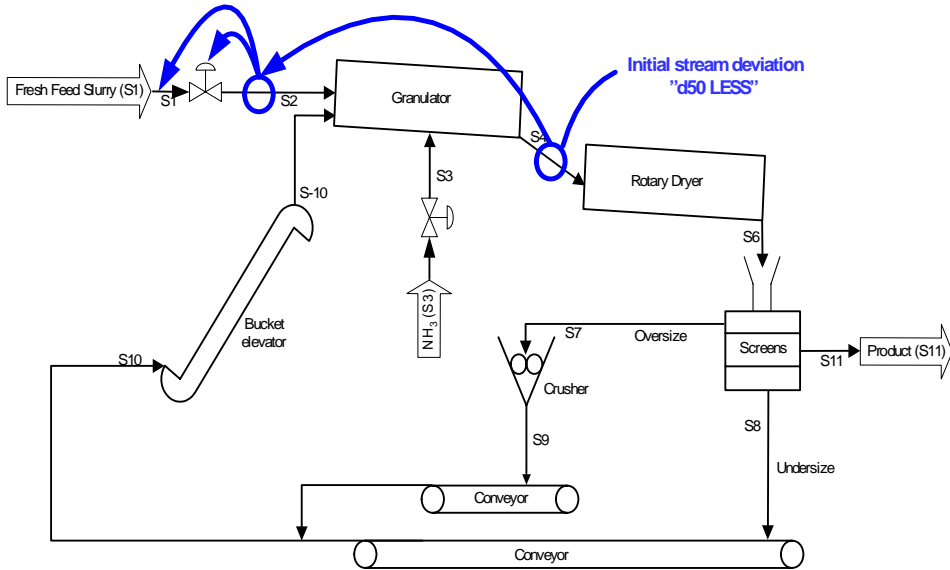


Figure 3: The flowsheet of the granulator circle and the steps of the diagnostic reasoning

A granulátor diagnosztikai rendszer tudáselemei

A granulátor diagnosztikai rendszerében kétféle típusú tudáselem található: a *dinamikus folyamatmodellek*, amelyek differenciál-algebrai egyenletek formájában tartalmazzák a folyamatrendszer hagyományos mérnöki ismereteit; valamint a szisztematikusan összegyűjtött *heurisztikus információk*, amelyek HAZOP és/vagy FMEA analízis eredményeiből származnak.

Az esettanulmányban felhasznált HAZOP tábla és FMEA tábla részeket a 4. ábra tartalmazza. A HAZOP tábla definiálja a (statikus) ok-következmény kapcsolatokat a szimptómák és a lehetséges okok között, amelyek felhasználásával logikai következtetésekkel eljuthatunk a deviancia gyökér okaihoz. Két egymással kapcsolatban álló szimptómát tartalmaz a HAZOP táblázat, ahol mindegyik szimptómához legalább két különböző ok tartozik. Egy lehetséges okot gyökér oknak tekintünk abban az esetben, amennyiben az a rendszer egy fizikai komponensének egy meghibásodási módjához kapcsolódik (ilyen például a HAZOP tábla második sorában a (2)-vel jelölt ok). Amikor egy ilyen gyökér okot talál a diagnosztikai rendszer, a diagnosztikai eredmény kiegészíthető vagy finomítható a megfelelő FMEA bejegyzésnek megfelelően, ahogy ez az FMEA táblázatban is látható.

4. ábra

HAZOP és FMEA táblákon végzett következtetés

Guide-word	Deviation	Possible causes	Consequences	Action required
Mean Particle Diameter (D50)	LESS	(1) Decrease in fresh feed size	* decrease in system holdup * change in granulation condition	a) increase fresh feed size b) change to original feed type c) increase slurry flow
		(2) Decrease/loss of slurry flow	* change in recycle PSD	
Slurry feed flow	LESS	(1) operator error in setting the flowrate	* reduced liquid phase in granulator	
		(2) failure in valve actuator	* lack of granulation	
		(3) failure in valve causing closure	* lower product size range flow from granulator	
		(4) reduced slurry production in preneutralizer		

The relevant part of the HAZOP table

Component	Description	Failure mode	Possible causes	Effects		Detection	Criticality	Action
				Local	System			
FCV	Slurry flow control valve	Stuck	maintenance failure	loss of flow control	potential product quality impacts	indirectly via product quality	MEDIUM - quality reduction in product	review maintenance procedures
		Closed	corrosion	lower or no flow	no growth			
		Open				D50 reduces in product		

The relevant part of the FMEA table

Figure 4: The HAZOP and FMEA table and the reasoning on them

Szimulációs eredmények

Az ismertett ágens-alapú diagnosztikai rendszer működésének illusztrálásához a rendszer egy részét, nevezetesen a diagnosztikai ágensek együttműködését mutatjuk be. Egy olyan esetet illusztrálunk, amelyben a diagnosztika eredménye különböző hibadetektálási és -izolálási módszerek kombinációjával kapható meg. Ezen ágens részrendszer szerkezete látható az 5. ábra bal oldalán. A bemutatott diagnosztikai ágensek fő viselkedése a heurisztikus tudáson (HAZOP, FMEA) alapuló logikai következtetések – ezt a JESS következtető gépe segíti –, amelyet egy folyamat szimuláló ágens egészít ki. Eltekintve a JADE beépített, főkonténerben (Main-Container) lévő ágenseitől, az ágens platform három konténeret tartalmaz:

- a *Real-Time Agents* nevű konténer a valós idejű ágensek számára (*MonitoringAgent* és *PreProcessorAgent*),
- a *Diagnostic Agents* nevű konténer a diagnosztikai ágensek számára (*SymptomGeneratorAgent*, *FaultIsolatorAgents* – ezek HAZOP és FMEA analízisen alapulnak –, *CompletenessCoordinatorAgent* és *LossPreventorAgent*)
- a *Process Agents* nevű konténer pedig egy folyamat ágens (*ProcessOutputPredictor*) tartalmaz.

5. ábra

Az ágens részrendszer struktúrája és a részrendszeren belüli kommunikáció

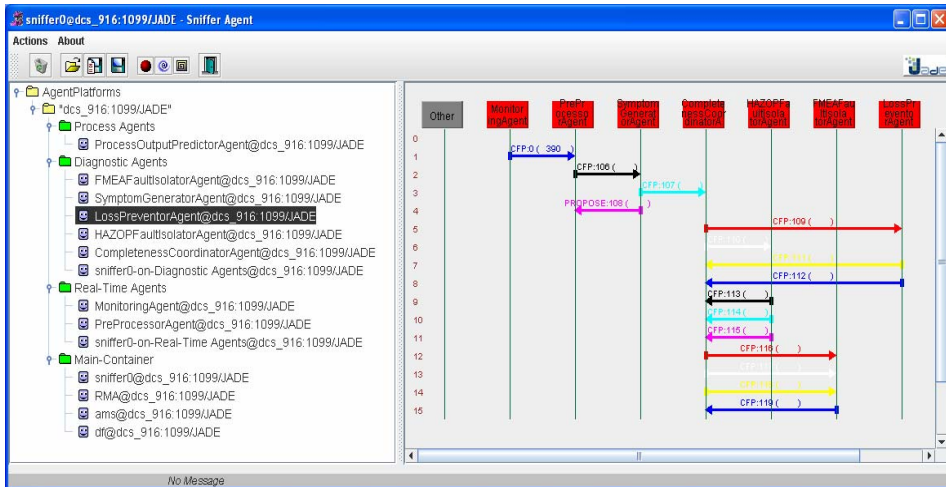


Figure 5: The structure and the communication of the agent system

Az ágensek közötti kommunikáció és működés egy részletét az 5. ábra jobb oldala szemlélteti. A hibadiagnosztika folyamatát a diagnosztikai ágensekkel azon szituációban mutatjuk be, amikor a granulátorból kilépő átlagos részecske átmérő (d_{50}) kisebb, mint egy megadott határérték. Ez az eset megfelel a 4. ábrán szereplő HAZOP tábla első sorának, amelyet a „Mean particle diameter (d_{50}) LESS” szimptóma ír le. Természetesen sok egyéb, ennél jóval bonyolultabb hiba is lehetséges egy ilyen ipari rendszerben.

A *MonitoringAgent* ágens által szolgáltatott változó értékei alapján a *PreProcessorAgent* ágens meghatározza a rendszerben lévő devianciákat. Abban az esetben, ha van detektált deviancia, a *SymptomGeneratorAgent* ágens ellenőrzi a szimptómák jelenlétét – jelen esetben a „ d_{50} LESS” szimptómát észleli –, és informálja erről a *CompletenessCoordinatorAgent* ágenszt. A *CompletenessCoordinatorAgent* ágens továbbítja a az észlelt szimptómát a *HAZOPFaultIsolatorAgent* és *LossPreventorAgent* ágenseknek, hogy meghatározzák a lehetséges hibákat és javasoljanak megelőző beavatkozásokat a HAZOP táblában szereplő információk felhasználásával. A *HAZOPFaultIsolatorAgent* ágens következtetés sorrendjét a 3. ábrán látható folyamatábra és a 4. ábrán látható HAZOP táblarészen szereplő jelölés szemlélteti. A HAZOP táblából látható, hogy a következtetés három lehetséges gyöker okot talál (azaz olyan okokat, amelyek a rendszer elemi komponenseinek meghibásodásaihoz kapcsolódnak), ahogy az a tábla második sorából kiolvasható.

A diagnosztikai és veszteség megelőző eredmények halmazát többszörös hibák esetén az FMEA analízisből származó táblán alapuló *FMEAFaultIsolatorAgent* ágens pontosítja, amelyet a *CompletenessCoordinatorAgent* ágens hív meg a HAZOP táblázatból származó információk alapján. Eredményként a „Slurry flow control valve fails Closed” gyöker ok hatását állítja elő („ D_{50} reduces in product”), amely azonos az észlelt szimptómával. A *LossPreventorAgent* ágens által meghatározott lehetséges

megelőző beavatkozások három lehetőséget írnak le a HAZOP tábla első sorának utolsó oszlopa szerint. A javaslatok alapján a *CompletenessCoordinatorAgent* ágens aktiválja a *ProcessOutputPredictor* ágenst, hogy megbecsültesse a rendszer jövőbeli viselkedését a megelőző beavatkozások segítségével.

6. ábra

A HAZOPFaultIsolatorAgent ágens következtetései egy része

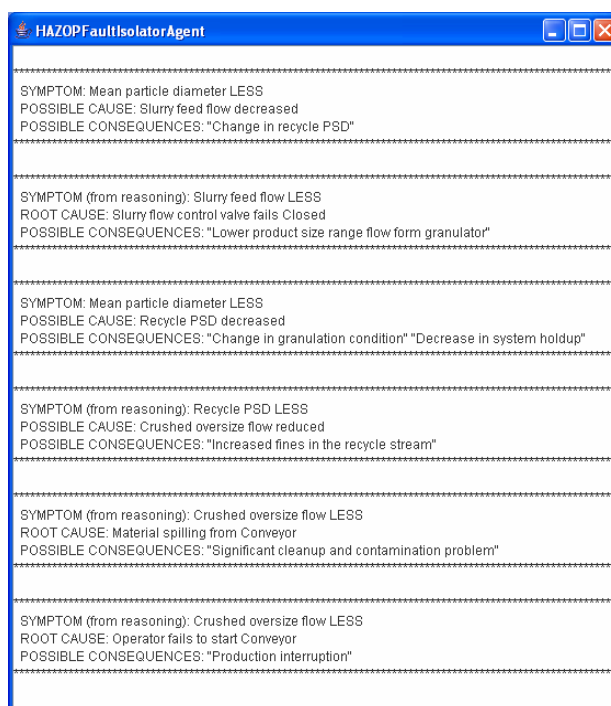


Figure 6: The HAZOPFaultIsolatorAgent's conclusion

A *HAZOPFaultIsolatorAgent* ágens következtetései egy részletét szemlélteti a 6. ábra, amelyen következtetés lépései – az észlelt és a logikai következtetés során előállított szimptómák a hozzá tartozó lehetséges okokkal illetve következményekkel – láthatóak.

KÖVETKEZTETÉSEK

Folyamatrendszerek diagnosztikai feladatainak megvalósítására alkalmas új módszer mutat be a cikk. A kifejlesztett multiágens-alapú diagnosztikai rendszer segítségével lehetővé válik a heterogén információforrásból származó hibadetektálási és -diagnosztikai technikák együttes kezelése, amely során a különböző diagnosztikai feladatokat megvalósító ágensok a koordináló ágensok közreműködésével folyamatosan finomítják egymás eredményeit, ezáltal kiküszöbölhetőek az egyes módszerek hátrányai, valamint ötvözhetőek a különböző módszerek előnyei.

KÖSZÖNETNYILVÁNÍTÁS

A szerzők köszönetüket fejezik ki az Országos Tudományos Kutatási Alapnak a T042710 és a T047198 számú kutatási támogatásért.

IRODALOM

- Blanke, M., Kinnaert, M., Junze, J., Staroswiecki, M., Schroder, J., Lunze, J. (2003). *Diagnosis and Fault-Tolerant Control*. Springer-Verlag.
- Cameron, I.T., Raman, R. (2005). *Process Systems Risk Management*. Elsevier.
- Cameron, I.T., Wang, F.Y., Immanuel, C.D., Stepanek, F. (2005). Process systems modelling and applications in granulation: A review. *Comput. Chem. Engng.*, 60. 3723–3750.
- Crawley, F., Tyler, B. (2000). *HAZOP: Guide to best practice*. The Institution of Chemical Engineers, Rugby, U.K.
- Federal Aviation Administration (2000). *System Safety Handbook*, chapter 9: Analysis Techniques.
- JADE - Java Agent DEvelopment Framework (2005). <http://jade.tilab.com>
- Jennings, N.R., Wooldridge, M.J. (1998). *Agent Technology*. Springer-Verlag, Berlin.
- JESS, the Rule Engine for the Java platform (2005). <http://herzberg.ca.sandia.gov/jess/>
- Lakner, R, Németh, E., Hantos, K.M., Cameron, I.T. (2006a). Agent-based diagnosis for granulation processes. *Computer-Aided Chemical Engineering*, 21B. Elsevier, (Eds.: W. Marquardt and C. Pantelides) 1443-1448.
- Lakner, R, Németh, E., Hantos, K.M., Cameron, I.T. (2006b). Multiagent realization of prediction-based diagnosis and loss prevention. *Lecture Notes in Computer Science*, 4031: *Lecture Notes in Artificial Intelligence*, Springer-Verlag, (Eds.: M. Ali and R. Dapoigny) 70-80.
- Patton, R. J., Frank, P. M., Clark, R. N. (1989). *Fault Diagnosis in Dynamic Systems: Theory and Applications*. Prentice-Hall, Englewood Cliffs, NJ, 47–153.
- The Protege Ontology Editor and Knowledge Acquisition System (2004). <http://protege.stanford.edu>
- Ungar, L.H., Venkatasubramanian, V. (1990). *Artificial intelligence in process systems engineering: knowledge representation*. CACHE, Austin, TX.
- Yang, A., Marquardt, W., Stalker, I., Fraga, E., Serra, M., Pinol, D. (2003). *Principles and informal specification of OntoCAPE*. Technical report, COGents project, WP2.

Levelezési cím (*Corresponding author*):

Németh Erzsébet

Magyar Tudományos Akadémia

Számítástechnikai és Automatizálási Kutató Intézet

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, POB. 63.

Tel.: 36-1-279 6163, Fax: 36-1-466 7503

e-mail: nemethe@sztaki.hu