

A CONFLET RENDSZER ÚJ ARCHITEKTÚRÁJA

Pasztuhov Dániel, dani@iit.bme.hu
Dr. Szeberényi Imre, szebi@iit.bme.hu
BME IIT

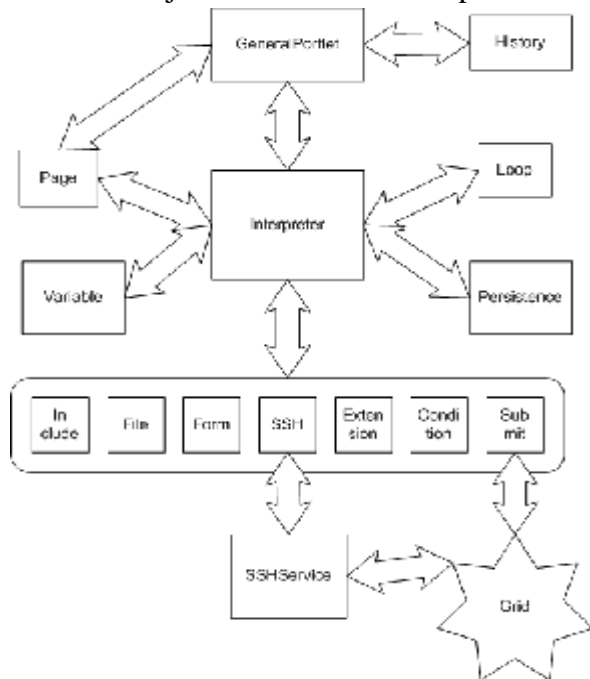
1 Bevezetés

A tavalyi Networkshop konferencián már bemutattuk a Confler rendszert [13, 14, 15], mely kifejlesztésének egyrészt az volt a célja, hogy a felhasználók olyan felületet kapjanak, amin a futtatókörnyezet sajátosságait figyelmen kívül hagyva indíthatnak felhasználóbarát felületen feladatot. Mindezt úgy, hogy a fejlesztőknek ne kelljen hasonló feladatok tömkelegét végrehajtani minden egyes feladathoz.

A Confler alapfilozófiája, hogy egy keretrendszert biztosít a fejlesztőknek, amelyet XML- és JSP- és egyéb állományok elkészítésével ér el. A leggyakoribb funkciók eléréséhez nem kell Java nyelven sem tudnia.

A Confler korábbi architektúrája egy a jelenleginél szűkebb funkciókészlethez volt tervezve, és amely az idők során egyre inkább idejétmúlttá vált. Bár – fejlesztői szemmel nézve – újabb parancsok hozzáadása nem volt bonyolult, kimondottan elegánsnak sem lehetett nevezni. Nagyban függött a – már egyre kevésbé fejlesztett – GridSphere Portál Keretrendszertől [10, 11, 12], valamint nem tudott ugyanazzal a szemmel nézni a különféle futtatókörnyezetekre, valamint a hibakezelése is elég szegényes volt.

Az új architektúra ezeket a problémákat küszöböli ki. A fő célkitűzés a minél nagyobb



1. ábra

fokú rugalmasság elérése volt, így jelenleg a parancsok hozzáadása, a portál keretrendszer lecserélése vagy új futtatókörnyezet hozzáadása sokkal egyszerűbb, mint korábban.

Cikkünkben először a régi architektúrát vizsgáljuk meg részletesebben (2. fejezet), majd az új architektúrával szemben támasztott követelményeket bontjuk ki részletesebben (3. fejezet). Ez után az új architektúra elemeit mutatjuk be részletesen (4. fejezet), majd a gyakorlati alkalmazás lehetőségeit tárgyaljuk, kitérve egy konkrét, ipari feladat követelményeire adott megoldásunk ismertetésére.

2 A régi architektúra

A Confler rendszer eredeti architektúrája az 1. ábrán látható. Bár moduláris felépítésű volt, a modulok granularitása nem volt megfelelő mértékű, és a modulok nem kellően függetlenek egymástól.

Bár a GeneralPortlet modul az

Interpretertől külön lett megvalósítva, az Interpreter szorosan össze van kapcsolva a GeneralPortlettel, annak GridSphere-függését is örökli, így a GridSphere-től való függetlenítés nagy nehézségekbe ütközne.

Az Interpreter – mint Confllet modul – meglehetősen monolitra sikerült, így minden egyes új parancs hozzáadása egy nagy osztály módosításával volt egyenértékű. Ráadásul a hibajelzés sem volt túl kifinomult. Az Interpreterben egyáltalán nem volt megoldva, hogy támogassa a különböző controller-verziókat.

További probléma volt, hogy a Confllet rendszer nem volt eléggé rugalmas a különféle middleware-ek kezelésében, így egy új middleware hozzáadáshoz mélyreható változásokra lett volna szükség.

3 Követelmények

Az új architektúrával szembeni követelményeket csoportokra bontottam:

- Futtatókörnyezet
 - Használja ki a JDK 5-ös verziójának lehetőségeit.
 - Használjon GridSphere Portál Keretrendszert, de legyen attól minél inkább független.
- Értelmező
 - Legyen az értelmező rész független a portál keretrendszertől lehetővé téve a külön fejlesztést, tesztelést.
 - Használható legyen több, különböző verziójú értelmező is.
 - Legyen lehetőség arra, hogy önálló Java alkalmazásként is fusson.
- Parancsok
 - Legyen könnyű új parancs hozzáadása.
 - A parancsok végrehajtása, kiértékelése a programozó oldaláról legyen egyszerű.
 - Legyen a parancsoknak visszatérési értékük, legyen megvalósítható egy olyan parancs, mely a visszatérési értéket egy változóba helyezi.
 - Legyen egyesítve a fülön való kattintással történő és a programozott oldalváltás kódja.
- Segédosztályok és köztesrétegek (middleware)
 - A rendszer legyen minél függetlenebb a köztesrétegek elérését szolgáló moduloktól.
 - Tudjon a rendszer middleware nélkül is működni (tesztelési céllal)
 - A middleware-elérő modulok legyenek lecserélhetők, hozzáadhatók, eltávolíthatók.
 - Middleware-modullá kell tenni a Condor elérését, és segédosztállyá a távoli parancsok végrehajtását célzó részt.
- Hibák kezelése
 - Minden hibáról adjon részletes tájékoztatást.
 - A statikusan kideríthető hibákat jelezze a konfiguráció feltöltésekor.
 - A futtatás közbeni hibák esetén a szöveges tájékoztatás mellett jelezze, hogy a hiba mely controller melyik részét állt elő (stack trace)

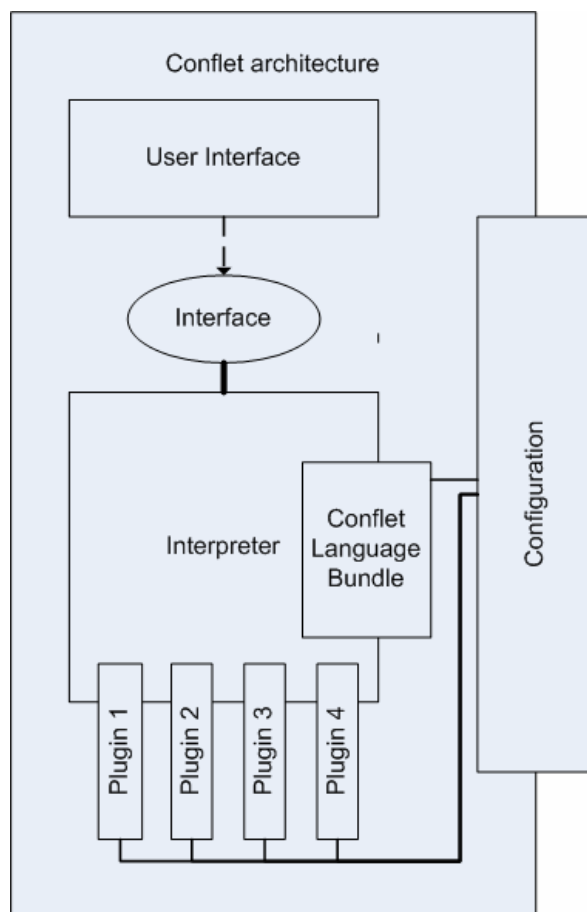
4 Az új architektúra

Az új architektúráról készült ábra a 2. ábrán látható. A nagy doboz jelképezi a Conftet teljes rendszerét, melybe a konfiguráció (Configuration) csak beilleszkedik, de kapcsolata van (használja) a Conftet különböző moduljait.

A legfelső egység a „User Interface”, mely csak lazán csatolódik a Conftet rendszer többi részéhez az Interface modulon keresztül. A Conftet két része (UI és a többi) egymással szembeni feltételezéseit az Interface modul testesíti meg.

Az Interpreter modul általános vezérlőmodul a rendszerben, de az attól függetlenül megvalósított Conftet Language Bundle (CLB) a parancsok és a konfigurációs fájlok megvalósításait tartalmazza.

A szintén az Interpreter részeként, de függetlenül megvalósított bővítmények (plugin) valósítják meg a külső rendszerekkel (middleware-ek, storage, távoli parancsfuttatás) való kapcsolatot.



2. ábra

4.1 User Interface

A User Interface modul fenntartja a kapcsolatot a Conftet rendszer fő része és a portál keretrendszer (vagy a felhasználói felület kezelésére használt – akár parancssoros Java-program) között. A user interface-nek a következő feladatai vannak:

- Felületet nyújt a rendszerszintű konfiguráció elvégzéséhez.
- Felületet nyújt a felhasználói konfiguráció elvégzéséhez.
- Kezeli a konfigurációs könyvtárakat (a CLB, a bővítmények, valamint a konfigurációs fájlok könyvtárait)
- Ha a felhasználó új feladatot akar indítani, elindítja a konfigurációs fájlok értelmezését (meghívja az Interpreters).
- Az Interpreter által kijelölt view fájlokat megjeleníti, kezeli a benne található bean-eket, előállítja a fülek területét.
- A felhasználói interakciót továbbítja az Interpreternek.
- A Conftet fő része és a portál keret-

rendszer által használt bean-eket átalakítja egymásba.

- Az Interpreter által nem kezelt eseményeket (a feladatindítás végrehajtása, az értelmezés lezárása stb) elkapja és lekezeli.
- Lekezeli az Interpreter által kiadott kérélmeket fájlok letöltésére (alkalmazáserververtől a felhasználó irányába).
- Kezeli az elindított feladatok listáját.

4.2 Interface

Az Interface modul feladata, hogy szabályozza a user interface modul és a Conftet többi részének interakcióját úgy, hogy mindkét rész a lehető legfüggetlenebb maradjon a másiktól. Segítségével a user interface modult megvalósító, portál keretrendszerrel függő modul

anélkül kicserélhető, hogy a Confllet többi részében bármit módosítani kellene. Hasonlóképpen, ha az Interpretert változtatnánk meg, az nem érintené a user interface modulját.

Az Interface modul a következő tulajdonságokkal rendelkezik:

- Interfészt nyújt a bővítmények inicializálásához.
- Általános interfészt nyújt a Confllet parancsaihoz anélkül, hogy azok pontosan definiálva lennének (ez a CLB-ben történik)
- Általános interfészt nyújt a Confllet által használt konfigurációs fájlok osztályaihoz anélkül, hogy pontosan meg lenne határozva, hogy milyen konfigurációs fájlok léteznek.
- Interfészt nyújt a konfigurációs fájlok betöltéséhez.
- Általános interfészt nyújt néhány konfigurációs fájlhoz (view, controller, tabs). Ezeket a user interface-nek is ismernie kell.
- Interfészt nyújt az user interface és a Confllet fő modulja közötti adatáramláshoz.
- Általános Interpreter-interfészt nyújt.
- Általános változótár-interfész definiál.
- Meghatározza az Interpreterben használt bean-ek osztályait.
- Funkcionalitást biztosít arra, hogy a rendelkezésre álló bővítmények közül a felhasználó kiválassza azt, amiket használni szeretne.
- Definiálja a kivételek osztályait.

4.3 Interpreter

Az Interpreter modulnak két fő feladata van a Conflleten belül. Az egyik az osztályok betöltése, a másik az értelmezés vezérlése.

4.3.1 Osztálybetöltés

A CLB és a bővítmények a fájlrendszerben JAR-fájlokként jelennek meg. Használatukhoz a JAR-fájlokból betöltött osztályokat (parancsok és konfigurációs fájlok osztályai) példányosítjuk, majd ezen kezdeti objektumok hoznak létre magukból újabb példányokat a fájlrendszerben található konfigurációs fájlok tulajdonságai alapján. A controller fájlok értelmezéséhez hasonló módszert használunk: A rendszer végigvizsgálja a controller fájlban található XML nyelvű parancsokat, a kezdeti objektumok segítségével értelmez minden egyes parancsot, és neki megfelelő objektumot hoz létre. Az objektumok hierarchiája megfelel a controller-ben lévő parancsok hierarchiájával.

4.3.2 Vezérlés

Az Interpreter kétfázisú inicializálást használ: rendszerszintű és normál inicializálást. Az oka, hogy a bővítmények kijelölése a magasabb szintű rétegek feladata, így a kiválasztáshoz külön interfészt kellett írni. Viszont a megfelelő bővítmények betöltését bizonyos inicializáló műveleteknek meg kell előzniük, más műveleteknek viszont később kell következni.

Az Interpreter modul ebben a rendszerben Façade tervezési mintát valósít meg.

4.4 Confllet Language Bundle

A CLB modulja főként a konkrét konfigurációs fájl-osztályok valamint a konkrét parancsosztályok tárolására hivatott. Ezen kívül definiálja a segédosztályok (távoli rendszer-elérés, helyi és távoli fájlkezelés, middleware-ek elérése) által megvalósított interfészeket. (A Conflletben csak ezen a szinten kerülnek elő konkrétan a segédosztályok interfészei, így azok módosítása mindössze a CLB módosítását jelenti – ez sokkal rugalmasabbá teszi az architektúrát).

4.5 Bővítmenyek

A bővítmenyek – mint már említettük korábban – a fájlrendszerben JAR-fájlokként jelennek meg. Tartalmazhatnak parancsokat – melyek egyesülnek a CLB parancsaival –, valamint tartalmazhatnak segédosztályokat a különféle rendszerekkel való kapcsolat lebonyolítására. Ezen kívül tartalmazhatnak még egy saját inicializálásukat végző osztályt is.

A bővítmenyek között függőségi relációk is definiálhatók (plugin.xml). Megadható minden bővítmenyhez, hogy egy megadott interfész (egy egyszerű karakterlánc) hányas verzióját valósítja meg, továbbá az is, hogy a megadott implementációnak mi a neve (szintén egy egyszerű karakterlánc – ami a bővítmenyek kiválasztásánál is megjelenik), és hányas verzió. A bővítmeny megadhatja, hogy mely interfészből legalább hányas és legfeljebb hányas verziót igényel. Ha ez rendelkezésre áll, a bővítmeny betöltődik, ha nem, akkor a betöltés elmarad.

5 A gyakorlati alkalmazás lehetőségei

A Confllet rendszer célja, hogy egyszerűbbé tegye a grid és párhuzamos feladatokat indító felhasználók dolgát azzal, hogy a feladat adatainak megfelelő felületet nyújt a felhasználónak kitöltésre. Mindezt úgy, hogy közben – egyúttal – a fejlesztők dolgát is leegyszerűsíti azzal, hogy igyekszik az ismétlődő feladatokat a fejlesztési folyamatból kiküszöbölni, ehhez egy paramétereztető keretrendszert biztosít a fejlesztőknek (vagyis konfigurátoroknak). A Confllet – célját túlszárnyalva – nem csak párhuzamos és grid feladatok indítására alkalmas, hanem bármilyen parancssoros alkalmazás elindítására képes.

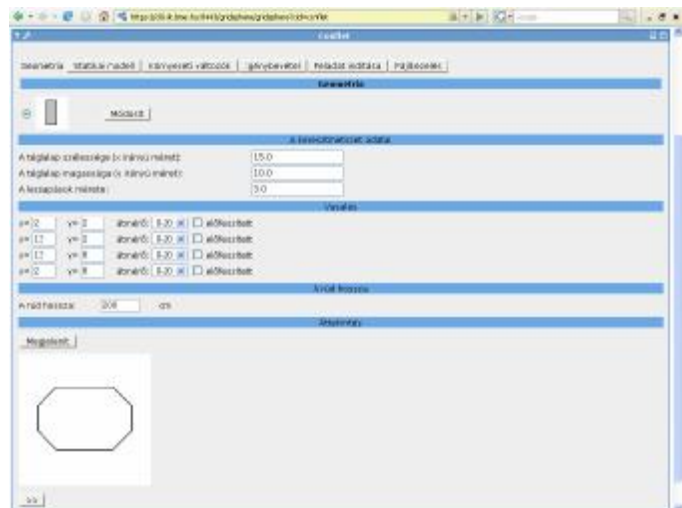
A Confllet rendszer lehetőségei a korábbi architektúrához képest az új architektúrával tovább bővültek. Az architektúra újdonságai számos ponton rugalmasabbá tették a rendszert, az tovább függetlenedett mind a kiszolgáló portál keretrendszertől, mind a feladatot elvégző grid vagy párhuzamos rendszertől. Újdonságai közül a következőket kell kiemelni:

- Könnyen átültethető a Confllet rendszer egyik portál keretrendszerről a másikra.
- Könnyen átültethető a feladat egyik middleware-ről a másikra, miközben lehetővé vált, hogy olyan parancsokat definiáljunk, melyek függetlenek a pillanatnyilag használt middleware-től.
- Gyorsabb lett, mivel indulás után betölti a lapokat (a hibakeresés miatt), így azokat nem kell a lapváltáskor újra átnézni, azok a memóriából töltődnek le.
- Könnyen adhatunk a keretrendszerünkhöz új parancsot. Mindössze egyetlen osztály megírására és rendszerbe illesztése a feladat.
- A korábbiaknál kiterjedtebb és alaposabb hibakezelés áll rendelkezésünkre, mely a szintaktikai és működés közbeni hibák felderítésére is alkalmas.

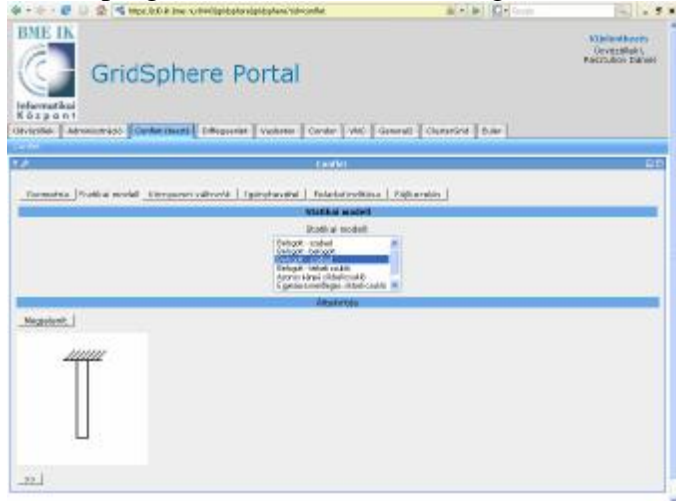
Bár a Confllet alkalmas grid, párhuzamos és parancssoros alkalmazás elindítására, kivéve azokat, melyek a futás későbbi részében felhasználói interakciót igényelnek. Ennek ellenére alkalmazhatósága igen széleskörű, hiszen a számítás-igényes problémák a fent említett osztályba sorolható.

6 Vasbeton hídgerenda-vizsgáló alkalmazás

A Confllet rendszer segítségével felhasználói felületet készítettünk egy építőipari probléma [5, 6, 7] megoldására.



hoz. A program előfeszített vasbeton gerendák [2, 4] és különösen nyomott oszlopok térbeli alakváltozását számolja a geometriai és anyagi nemlinearitások figyelembe vételével. A térbeli alakváltozások a görbület rúd tengely mentén történő integrálásával számíthatók. A rúd két végének megfogási viszonyai adják a feladat peremfeltételeit. Az algoritmus magja egy globálisan konvergens rekurzió, amely képes meghatározni a különösen nyomott rúdkeresztmetszethez tartozó görbületet. Az iteratív eljárást [1, 3] a peremérték-feladatok megoldására szolgáló Párhuzamos Hibrid Algoritmusba [8, 9] ágyaztuk be.



Habár a javasolt módszer nagyon robusztus, a kielégítően pontos számítás csak párhuzamos környezetben valósítható meg, mivel legalább egymillió rúdalak meghatározása szükséges.

A cél az volt, hogy olyan felhasználói felületet hozzunk létre, mellyel a módszer elérhetővé válik az ipari felhasználók számára is. Egy, a mérnöki életből vett gerenda vagy oszlop leírása 200-1000 paramétert igényel, ezek száma a geometriai komplexitástól függ. A paraméterek többsége a betonkeresztmetszet geometriáját, a vasbetétek és/vagy elfeszítő pászmák helyét és keresztmetszeti területét írja le. A felhasználó által bevitt keresztmetszeti adatok a geometria ellenőrzése céljából grafikusan megjelenítendők. Szintén meg kell adni és meg kell jeleníteni a rúd két végpontjához tartozó, a mérnöki gyakorlatban előforduló és mechanikailag lehetséges megfogás-kombinációkat. Az anyagi tulajdonságoknak meg kell felelniük az Eurocode2 szabvány előírásainak. A felhasználó különböző, a szabvány által megadott szilárdságú beton, betonvas és előfeszítő pászma közül választhat. A kiválasztott anyagminőséghez a felhasználói felületnek társítania kell néhány egyéb anyagjellemzőt is, például a rugalmassági moduluszt. További paraméterek a rúd hosszát, a relatív páratartalmat, a gerenda életkorát az első megterhelés időpontjában, az esetleges hőkezelés hosszát és hőmérsékletét stb. írják le. További paraméterekkel adhatók meg a rúd terhei, lehetőség van megoszló és koncentrált terhek bevitelére két, egymásra merőleges irányban. Az összes paraméter megadása után a felület elindítja a feladatot a megadott környezetben. Az elkészült alkalmazás – a Confler új architektúrájának köszönhetően – könnyedén átül-tethető egyik környezetből a másikba.



7 Köszönetnyilvánítás

E munka részben a Nemzeti Kutatási és Technológiai Hivatal Pázmány Péter programjának (RET-06/2005) támogatásával jött létre. A szerzők szeretnék kifejezni a köszönetüket az EGEE projektnek is (EU INFSO-RI-031688)

8 Összefoglalás

Cikkünkben bemutattuk a Confllet rendszer új architektúráját, mellyel a keretrendszer könnyebben bővíthető, rugalmasabb, mint elődje. Egyszerűbb új parancsokat hozzáadni, lecserélni a felhasználói felületet biztosító portál keretrendszert (lehet az akár szöveges felületű is), könnyebb új futtatókörnyezetet hozzáadni a rendszerhez. Kifinomultabb lett a hibakezelés is. Bemutattuk továbbá, hogy milyen felhasználási lehetőségei vannak a Conflletnek – mindezt egy konkrét, iparban is használt feladat kapcsán.

9 Hivatkozások

- [1] Domokos, G. & Gáspár, Zs. “A global, direct algorithm for path-following and active static control of elastic bar structures”, *Int. J. Struct. Mech.*, 1995;23, 549-571.
- [2] Sipos, A. A., Domokos G. “Asymmetrical, spatial deformations of reinforced concrete columns and prestressed beams”, *fib Symposium “Keep Concrete Attractive”*, 2005, Budapest, Vol. II, pp. 693-698.
- [3] Sipos, A. A., Domokos G., Gáspár Zs. “The convergence features of the 2D Pelikan iteration”, *J. of Building Science*, 2005, 33 (1-2), 205-217 (in Hungarian)
- [4] Domokos, G. “Global description of elastic bars”, *Zeitschr. Angew. Math. Mech.*, 1994;74, T289-T291.
- [5] Brøndum-Nielsen, T. “Stress Analysis of Concrete Sections Under Service Load”, *ACI Journal, Proceedings*, 1979, V. 76., No. 2, 195-211.
- [6] Brøndum-Nielsen, T. “Serviceability Limit State Analysis of Cracked, Polygonal Concrete Sections Under Biaxial or Symmetric Bending”, *ACI Journal, Proceedings*, 1986, V. 83., No. 2, 209-218.
- [7] Cosenza, E. and Debenardi, P. G. “Calculation of Stresses, Deformations and Deflections of Reinforced and Prestressed Concrete Elements in Service”, *CEB Bulletin 235*, 1997, 105-142.
- [8] Gáspár, Zs., Domokos, G., and Szeberényi, I. “A parallel algorithm for the global computation of elastic bar structures”, *Comput. Assist. Mech. Eng. Sci.* 1997;4, 55-68.
- [9] Domokos G., Szeberényi I. “A Hybrid Parallel Approach to One-parameter Nonlinear Boundary Value Problems”, *Comput. Assist. Mech. Eng. Sci.* 2004;11,15-34.
- [10] M. Russell, J. Novotny, O. Wehrens, “GridSphere: An Advanced Portal Framework”, *GridSphere Project Website (www.gridsphere.org)*
- [11] M. Russell, J. Novotny, O. Wehrens, “GridSphere: A Portal Framework for Building Collaborations”, *GridSphere Project Website (www.gridsphere.org)*
- [12] M. Russell, J. Novotny, O. Wehrens, “The Grid Portlets Web Application: A Grid Portal Framework”, *GridSphere Project Website (www.gridsphere.org)*
- [13] Pasztuhov Dániel, *Paraméterezhető portletok fejlesztése ClusterGRID Portál környezetben*, TDK dolgozat, Budapest, 2004.
- [14] Pasztuhov Dániel, *Konfigurálható felületű portletok fejlesztése és alkalmazása ipari feladatok megoldásában*, Diplomaterv, Budapest, 2005.
- [15] Pasztuhov Dániel, dr. Szeberényi Imre, „Konfigurálható portlet (Confllet) alkalmazása ClusterGrid környezetben”, *Networkshop 2006. konferencia*, Miskolc, <https://nws.niif.hu/ncd2006/docs/ehu/015.pdf>