

Azonnali üzenetküldés SIP protokollal

MUHI DÁNIEL

Pannon Egyetem Műszaki Informatikai Kar, Információs Rendszerek Tanszék
dani@best.vein.hu

Kulcsszavak: SIP, azonnali üzenetküldés, jelenlét, jelzés

A Session Initiation Protocol (SIP) olyan általános célú alkalmazási szintű internetes protokoll, melynek segítségével viszonyokat hozhatunk létre két vagy több felhasználó között. Ezek a viszonyok leggyakrabban internetes telefonhívások, valamint hagyományos vagy multimédiás konferenciák. A protokoll tervezése során fontos szempont volt a modularitás és a bővíthetőség. Ezáltal számos olyan szolgáltatás hozható létre segítségével, melyekre eredetileg fel sem készítették. Ebben a cikkben bemutatjuk magát a protokollt, illetve azokat a bővítéseket, melyek alkalmassá teszik egy azonnali üzenetküldő (instant messaging) rendszer megvalósítására.

1. Bevezetés

Az azonnali üzenetküldő (instant messaging) szolgáltatások rendkívül népszerűvé váltak az elmúlt évek során, egyes rendszerek felhasználóinak száma négyszázmillióhoz közelít. A legismertebb azonnali üzenetküldő rendszerek az ICQ, az AOL és a Yahoo! Instant Messenger, valamint az MSN Messenger. Jelenleg mindegyikük egymástól független és egymással nem együttműködő protokollokat használ azért, hogy megtartsa felhasználói bázisát. A protokollok működését nem publikálják, így nincs lehetőség olyan üzenetküldő alkalmazás kifejlesztésére, mely együttműködik a már létező rendszerekkel. Ezt felismerve az IETF létrehozta a *SIP for Instant Messaging and Presence Leveraging Extensions* (SIMPLE) nevű munkacsoportot, melynek feladata egységes üzenetküldés megvalósítása a SIP protokoll segítségével [1].

A SIP bővíthetősége ideálissá teszi erre a feladatra. Ezért a SIMPLE által kifejlesztett technológia valószínűleg vezető szerephez jut a különböző szolgáltatásnyújtók közötti üzenetátvitel szabványosításában. Ennek egyik jele, hogy a Microsoft és a Yahoo! 2005. október 13-án bejelentette, hogy az idei nyárra megvalósul azonnali üzenetküldő rendszereik közötti együttműködés. A megvalósítás a SIP/SIMPLE segítségével történik.

Az azonnali üzenetküldő rendszerekben az üzenetküldés mellett biztosítani kell a *jelenléti információt* (presence information). A SIP részben biztosítja a jelenléti szolgáltatást, de az azonnali üzenetküldéshez kiterjesztéseket (extensions) kell használni, melyeket e cikk mutat be.

2. A SIP protokoll

Az internetes telefonálás leginkább abban különbözik az egyszerű multimédia-szolgáltatásoktól, hogy viszonyokat használ a kommunikáció során. A viszonyok lét-

rehozását és kezelését *jelzésnek* (signaling) nevezzük. A két legfontosabb internetes jelzésrendszer az Internet Engineering Task Force (IETF) által kifejlesztett *Session Initiation Protocol* (SIP) és a *H.323*, mely az International Telecommunications Union (ITU) ajánlása.

A SIP kliens-szerver protokoll, azaz a kliens kéréseket küld a szervernek, mely feldolgozza azokat [2]. Mivel telefonáláskor bármely fél küldhet és fogadhat kéréseket, ezért a SIP-et használó rendszerek minden felhasználói oldalon tartalmazzák a protokoll kliens-, ill. szerver részét. Ezt a kettős viselkedésű elemet *SIP telefonnak* (user agent server) hívják. Egy másik protokollelem a proxy szerver, mely kéréseket fogad, és azokat továbbküldi egy másik proxy szervernek, vagy egy SIP telefonnak. Az *átírányítási szerver* (redirect server) feladata értesíteni a címzettet arról, hogy közvetlenül felveheti a kapcsolatot a hívóval.

A fenti három elem között csak a funkciókban van különbség: a proxy vagy átírányítási szerver nem fogadhatja vagy utasíthatja vissza a kéréseket, csak a SIP telefonnak van ehhez joga. Ez a modell hasonló a HTTP-hez, ahol ugyanaz a hoszt viselkedhet kliensként vagy szervertként is. Ugyanakkor a HTTP-ben is létezik proxy szerver, melynek funkciója hasonló a SIP-proxyhoz.

A SIP és a HTTP között van még egy hasonlóság: a SIP-ben használt üzenetek és fejrészek szintaktikája nagyjából azonos a HTTP/1.1-ben használtakkal. Azonban a SIP nem a HTTP kiterjesztése.

Egy SIP üzenet kétféle típusú lehet: vagy kérés (a szerverhez) vagy válasz (a kliensnek). Az üzenetek szöveges formájúak, és tartalmazznak egy *kezdősort* (start-line), egy vagy több *fejrészt* (header), és egy *opcionális szövegrészt* (message body). Ugyanúgy, mint a HTTP esetén, a kliens kérései valamilyen metódust aktiválnak a szerveren. A metódust a kezdősorban kell megadni, míg a fejrészek további információt tartalmazznak: például az üzenet küldőjének címét, az elküldés dátumát. A SIP-ben számos, a HTTP-ből ismert fejrész megtalál-

ható, pl. az entitás-fejrészek (Content-type), valamint a hitelesítők. Ez lehetővé teszi már létező kódok újrafelhasználását, valamint leegyszerűsíti a webszerverekkel való integrációt.

A SIP-ben számos metódust definiáltak. Ha egy felhasználóval kapcsolatot szeretnénk létesíteni, az **INVITE** metódust kell használnunk. A kapcsolat létrejöttét a hívó egy **ACK** üzenettel jelzi. Ha kapcsolatot szeretnénk létrehozni egy felhasználóval, használhatjuk előtte az **OPTIONS** üzenetet is, melyre a hívott fél tájékoztatást ad képességeiről, de maga a kapcsolat nem jön létre. A létrejött kapcsolatot bármely fél megszüntetheti egy **BYE** üzenettel. További metódus a **CANCEL**, mely egy hívásfelépítési folyamatot szüntet meg, de a már létrejött kapcsolatokra nincs hatással. Egy másik metódus a **REGISTER**, mely segítségével a kliens elindulása után bejegyezheti egy SIP-szerverre, hogy éppen hol érhető el.

A kapcsolat létrehozásához egyértelműen azonosítani kell a felhasználókat. Mivel az Interneten a legelterjedtebb címzési mód az e-mail cím, ezért a SIP-azonosítók is `user@domain` formájúak.

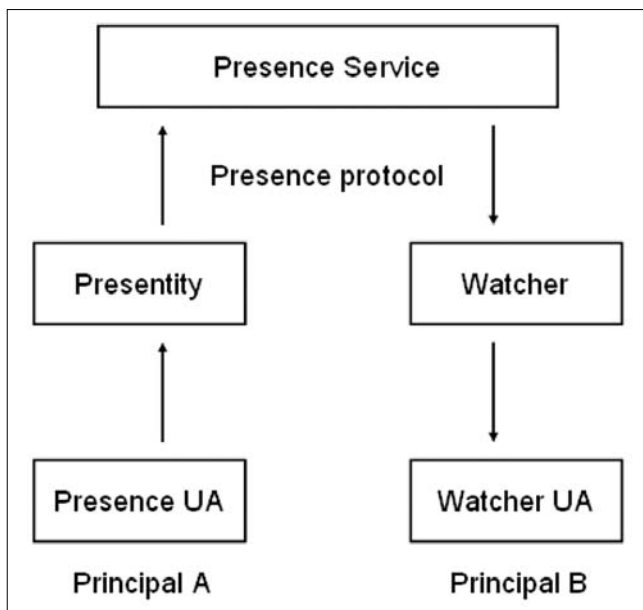
Ezeket a címeket a protokoll beágyazza az úgynevezett *SIP URL*-be, melynek szintaxisa: `sip:azonosító` (például `sip:somebody@example.com`).

3. Jelenléti és azonnali üzenetküldő szolgáltatások modellje

Az IETF egységes modellbe foglalta e két szolgáltatást, melyet a 2778-as RFC-ben publikáltak [3]. A szolgáltatások neve *Presence Service* és *Instant Message Service*. Az előbbi a jelenléti szolgáltatás központi eleme. Ez fogadja, tárolja és továbbítja a felhasználók állapotát. Az *Instant Message Service* a felhasználók egymásnak küldött üzeneteit fogadja és továbbítja.

A modell az 1. ábrán látható:

1. ábra Az IETF modell



A jelenléti szolgáltatás segítségével tudhatjuk meg egy távoli erőforrás állapotát. Ez az erőforrás leggyakrabban egy személyt jelent, akivel kommunikálni szeretnénk. A szolgáltatás két részből áll. Először az erőforrás (az ábrán az A személy) meghirdeti az állapotát.

Minden erőforrás kapcsolódik egy *Presence User Agent* (PUA)-hez, mely mindig a helyi gépen található (pl. az üzenetküldő szoftver). Ennek adja meg aktuális állapotát. A PUA továbbítja ezt az információt (*Presence Information*) a *Presentity*-hez, melynek feladata ezt eljuttatni a központi szerepet betöltő *Presence Service*-hez. A modellben csak a világosság kedvéért választották szét a PUA és a *Presentity* elemeket, ezek gyakran ugyanazon az eszközön helyezkednek el.

Ha a bejegyzés megtörtént, utána bármikor lekérdezhetjük A állapotát. A lekérdezési folyamat nagyjából a bejegyzés „tükörképe”. Ha B személy kíváncsi A állapotára, a *Watcher User Agent* (WUA)-hez kell fordulnia, mely – a PUA-hoz hasonlóan – szintén a helyi gépen van. A WUA számára a *Watcher* szerzi meg a kívánt információt a *Presence Service*-től. A WUA-t és a *Watcher*-t itt is csak funkciójuk miatt választották szét, hiszen általában ugyanazon az eszközön helyezkednek el. A rendszeren kívüli szereplők neve *Principal*. Ezek azok a felhasználók vagy szoftverek, melyek egymás állapotára kíváncsiak.

Az egyszerű lekérdezésen túl az is elképzelhető, hogy B lekérdezi A állapotát, majd azt kéri a *Presence Service*-től, hogy értesítse, ha megváltozik ez az állapot. Ez alapján a *Watcher*-eknek két típusát különböztethetjük meg: a *Fetcher* egyszerűen lekérdezi az állapotot, a *Subscriber* pedig lekérdezi az állapotot, és a továbbiakban tájékoztatást vár az állapotváltozásokról. A *Fetcher*-nek van még egy speciális típusa, a *Poller*, mely szabályos időközönként kérdezi le az állapotot.

A *Presence Information Presence Tuple*-ökből (PT) áll. Egy PT-ben szerepel az erőforrás állapota (status), melynek értéke lehet *open* vagy *closed*, továbbá tetszőleges állapotokat definiálhatunk, pl. *busy*, *away*. A PT többi eleme opcionális (a felhasználó címe, egyéb megjegyzés).

4. Értesítési funkció a SIP-ben

A SIP rendelkezik alapvető jelenléti funkciókkal. Ha felhívunk egy SIP telefont, akkor az a válaszüzenetben jelzi a felhasználó állapotát. Így például egy *200 OK* válasz esetén biztosak lehetünk abban, hogy a felhasználó online állapotban van. Ugyanakkor a *480 Temporarily Unavailable* vagy a *486 Busy Here* jelentheti azt, hogy a hívott kikapcsolt állapotban van, vagy pedig be van kapcsolva, de éppen nem tudja fogadni a hívást. Mindenesetre a SIP válaszüzenetek világosan megkülönböztetik a felhasználó online és offline állapotát.

Mindez azonban csak egyszeri lekérdezés, a hívónak nincs lehetősége arra, hogy értesüljön a hívott állapotának megváltozásáról. Ráadásul minden lekérdezés esetén fel kell hívni azt a személyt, akinek állapotára kíváncsiak vagyunk.

A probléma megoldására született egy javaslat [4], melynek célja a SIP kibővítése értesítési funkcióval. Ez azt jelenti, hogy egy SIP-kliens kérheti egy SIP-szervertől, hogy adott események bekövetkeztekor értesítést kapjon.

A modell legfontosabb elemei a Subscriber és a Notifier. A javaslat készítője két új metódust definiált: a **SUBSCRIBE**-ot és a **NOTIFY**-t. A SUBSCRIBE az INVITE-hoz hasonlít. A SUBSCRIBE segítségével kérdezhetjük le egy távoli erőforrás állapotát, illetve ezzel kérhetünk értesítést az állapot megváltozásakor.

A SUBSCRIBE üzenet *Request URI*-jában szerepel, hogy a Subscriber mely erőforrás állapotára kíváncsi, az *Event* fejrész pedig egy *eseménycsomag* nevét tartalmazza. Az eseménycsomag állapotinformációk olyan halmaza, melyek bekövetkezése esetén a Notifier-nek értesítést kell küdeni. Ezenkívül a Subscriber-nek az *Expires* fejrészben meg kell adnia egy időtartamot másodpercekben, mely a feliratkozás érvényességét határozza meg: az időtartam letelte után a feliratkozás érvényét veszti, hacsak közben nem frissítették. Ha az időtartamot 0-nak adjuk meg, ez leiratkozást jelent.

Ha a SUBSCRIBE kérés megérkezett a Notifier-hez, akkor az feldolgozza és létrehozza az előfizetést, valamint küld egy *200 OK* választ és egy NOTIFY üzenetet a Subscriber-nek. Ha valamiért nem tudja azonnal létrehozni az előfizetést (pl. a felhasználóra kell várnia), akkor egy *202 Accepted* választ küld a Subscriber-nek. A NOTIFY üzenet szövegrészét a SUBSCRIBE üzenet Accept fejrészében vagy az eseménycsomagban megadottak szerint kell formázni. Ez a szövegrész fogja tartalmazni az erőforrás állapotát, vagy egy URI-t, mely erre az állapotra mutat.

5. Jelenlét a SIP-ben

Mivel a SIP szerverek már eleve rendelkeznek a felhasználók állapotának adataival, ezért a SIP különösen alkalmas a jelenléti funkció megvalósítására. Továbbá, mivel a SIP hálózatok az INVITE üzeneteket mindig továbbítják ahhoz a proxy-hoz, mely tárolja a keresett felhasználó elérhetőségét, ezért a SUBSCRIBE üzeneteket is ugyanígy, a megfelelő proxy-khoz kell továbbítani. Ez azt jelenti, hogy a SIP hálózatokat egyszerűen felhasználhatjuk jelenléti szolgáltatások létrehozására.

A SIP értesítési funkciójának alkalmazásával már könnyen megvalósítható a presence szolgáltatás. A Session Initiation Protocol (SIP) Extensions for Presence draft [5] javaslat arra, hogyan lehetne ezt az általános funkciót jelenléti szolgáltatásra alkalmazni.

A dokumentum teljes mértékben figyelembe veszi az RFC2778-ban leírt architektúrát, és egy eseménycsomagot definiál, mely a *presence agent* fogalmára épül. A *Presence Agent* (PA) olyan User Agent elem, mely képes SUBSCRIBE üzenetek fogadására és megválaszolására, és ha bekövetkezik a kívánt esemény, értesítést küld róla a Subscriber-nek. A PA logikai entitás, mert általában más entitásokkal együtt helyezkedik el.

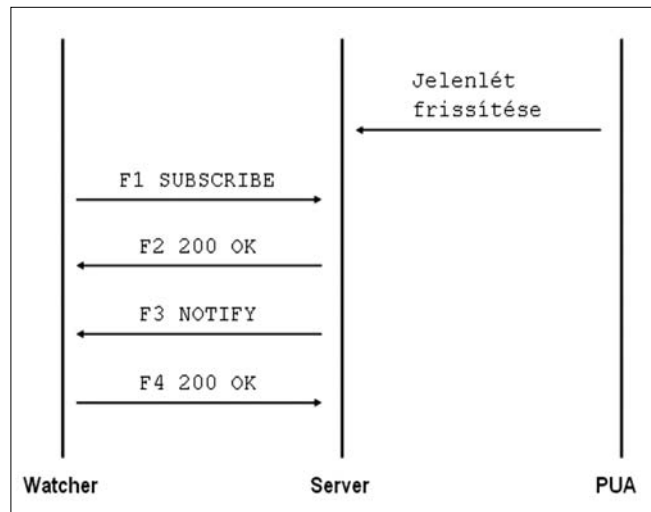
A *Presence Server* olyan fizikai entitás, mely PA-ként vagy proxy-ként viselkedhet. Ha proxy-ként viselkedik, akkor nem válaszol a SUBSCRIBE kérésekre, hanem továbbítja azokat egy PA felé.

Az előfizetett erőforrás állapotát a NOTIFY üzenet szövegrésze fogja tartalmazni. Az állapotleírásra definiáltak egy újfajta MIME típust, az *application/cpim-pidf+xml*-t [6]. Ezt a formátumot az IETF azzal a szándékkal, hozta létre, hogy a különböző jelenléti rendszerek közötti átjárást biztosítsa. Mivel a jelenléti információ hierarchikus szerkezetű, valamint könnyen bővíthetőnek kell lennie, ezért a PIDF XML-t használ az adatok tárolására.

Nézzünk egy egyszerű példát! Ebben az esetben a Watcher szeretne értesítést kapni egy erőforrás állapotról, mely a Presence User Agent (PUA)-en keresztül kapcsolódik a jelenléti rendszerhez. Feltételezzük, hogy az erőforrás már elküldte állapotát a szervernek.

Ebben az esetben a kommunikáció folyamata a 2. ábrán látható:

2. ábra Kommunikáció a SIP-ben



1) Először is a Watcher, melynek azonosítója *sip:user@example.com*, fel szeretne iratkozni a *sip:resource@example.com* azonosítójú erőforrás állapotára. Ehhez elküld egy SUBSCRIBE üzenetet, melynek Request URI-ja tartalmazza az erőforrás azonosítóját. A To, From és Call-ID mezők értelemszerűen lesznek kitöltve. Az Event értéke csak *presence* lehet, hiszen presence szolgáltatásról van szó. A Contact mezőben szerepel, hogy a szerver kinek küldje az értesítést. Ebben az esetben azt szeretnénk, hogy a bejegyzés 10 percig legyen érvényes, ezért az Expires értéke 600 másodperc lesz.

```

SUBSCRIBE sip:resource@example.com SIP/2.0
To: <sip:resource@example.com>
From: <sip:user@example.com>
Call-ID: 2010@watcherhost.example.com
Event: presence
Contact: <sip:user@watcherhost.example.com>
Expires: 600
  
```

2) Ha a szerver megkapta az üzenetet, azonnal visszaküld a Watcher-nek egy 200 OK választ, jelezve, hogy minden rendben van:

```
SIP/2.0 200 OK
To: <sip:resource@example.com>
From: <sip:user@example.com>
Call-ID: 2010@watcherhost.example.com
Event: presence
Contact: <sip:server.example.com>
Expires: 600
```

3) Ezután tájékoztatást küld a Contact-ban megadott címre a kívánt erőforrás állapotáról. Az üzenet szövegrészében egy PIDF dokumentum tartalmazza az állapotinformációt:

```
NOTIFY sip:user@watcherhost.example.com
SIP/2.0
From: <sip:resource@example.com>
To: <sip:user@example.com>
Call-ID: 2010@watcherhost.example.com
Event: presence
Subscription-State: active;expires=599
[PIDF document]
```

4) Ezt az üzenetet a Watcher nyugtázza:

```
SIP/2.0 200 OK
From: <sip:resource@example.com>
To: <sip:user@example.com>
Call-ID: 2010@watcherhost.example.com
```

6. Azonnali üzenetküldés

Azonnali üzenetküldés esetén a felhasználók majdnem valós idejű kommunikációt folytatnak egymással rövid üzenetek segítségével. A rövid üzenetek lehetővé teszik a gyors átvitelt, így párbeszéd jöhet létre a két fél között. Az azonnali üzenetküldés legismertebb példája az SMS, melyet tipikusan nem használnak párbeszédés módokban, mert erre nehézkes és költséges lenne.

A SIP-ben az azonnali üzenetküldést a MESSAGE kiterjesztéssel oldották meg. A kérés szövegrésze tartalmazza az elküldendő üzenetet. Az üzenet bármilyen MIME típusú lehet. A kérésre a küldő ugyanúgy választ kap, mint bármely más SIP kérés esetén. Ha az üzenet megérkezett, a fogadó ezt 200 OK válasszal nyugtázza. Ez a válasz nem feltétlenül jelenti azt, hogy a felhasználó el is olvasta az üzenetet.

A 3. ábrán látható, amikor egy felhasználó üzenetet küld egy másikkal proxy-n keresztül. Mindkét felhasználó a „domain.com” domain-ben helyezkedik el.

3. ábra
Üzenetküldés proxy-n keresztül

Nézzük meg, hogyan épülnek fel a kérések és a válaszok a példában!

1) Először is az első felhasználó elküld egy kérést a proxy-nak.

A kérés szövegrészében szerepel az üzenet egyszerű szöveggént. Ezt a Content-Type mező jelzi. A Content-Length az üzenet hosszát adja meg. A Max-Forwards értéke azt adja meg, hogy legfeljebb hány közbeeső hálózati elemen keresztül továbbítható a kérés:

```
MESSAGE sip:user2@domain.com SIP/2.0
Max-Forwards: 70
From: sip:user1@domain.com;tag=49583
To: sip:user2@domain.com
Call-ID: asd88asd77a@1.2.3.4
Content-Type: text/plain
Content-Length: 18
```

Watson, come here.

2) A proxy felismeri, hogy a címzett ugyanabban a domain-ben van, csökkenti a Max-Forwards mező értékét és tovább küldi a kérést:

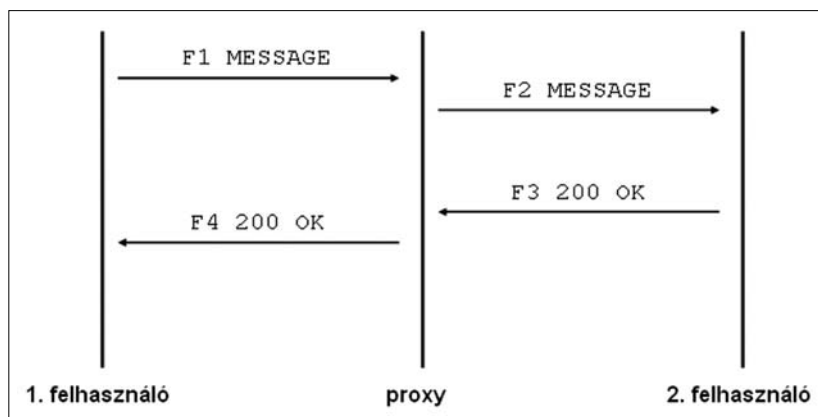
```
MESSAGE sip:user2@domain.com SIP/2.0
Max-Forwards: 69
From: sip:user1@domain.com;tag=49394
To: sip:user2@domain.com
Call-ID: asd88asd77a@1.2.3.4
Content-Type: text/plain
Content-Length: 18
```

Watson, come here.

3) Miután a fogadó alkalmazás megkapta az üzenetet, megjeleníti, és visszaküldi a proxy-nak a választ:

```
SIP/2.0 200 OK
From: sip:user1@domain.com;tag=49394
To: sip:user2@domain.com;tag=ab8asd9
Call-ID: asd88asd77a@1.2.3.4
Content-Length: 0
```

4) A proxy az előző választ küldi vissza az első felhasználónak.



7. Összefoglalás

Az azonnali üzenetküldő rendszerek közötti együttműködésnek több lehetséges módszere van, és úgy tűnik, hogy jelenleg ezek közül a legnépszerűbb az IETF által létrehozott SIP alapú jelenléti információs és azonnali üzenetküldési modell. A modell három legnagyobb erőssége az, hogy egyszerűen megvalósítható, szabványos és szabadon felhasználható.

E tulajdonságok miatt használják egyre többen a SIP/SIMPLE modellt azonnali üzenetküldő rendszerek integrálására.

Irodalom

- [1] SIP for Instant Messaging and Presence Leveraging Extensions (simple) IETF munkacsoport honlapja: <http://www.ietf.org/html.charters/simple-charter.html>
- [2] H. Schulzrinne, J. Rosenberg, „Internet Telephony: architecture and protocols – an IETF perspective”, Computer Networks 31 (1999), pp.237–255.
- [3] M. Day, J. Rosenberg, H. Sugano: A Model for Presence and Instant Messaging, RFC2778.
- [4] A. B. Roach: Session Initiation Protocol (SIP) – Specific Event Notification, RFC3265.
- [5] J. Rosenberg, D. Willis, H. Schulzrinne, C. Huitema, B. Aboba, D. Gurle, D. Oran: Session Initiation Protocol extensions for Presence, draft-ietf-simple-presence-07.
- [6] G. Klyne, D. Atkins: Common Presence and Instant Messaging (CPIM): Message Format, RFC3862.



Világszínvonalú elektronikai tervezőlaboratórium a BME-n

A Mentor Graphics Corporation, az elektronikai hardver- és szoftvertervező megoldások világpiaci vezetője szeptemberben a cég által szponzorált elektronikai tervezőlaboratóriumot létesített a BME Villamosmérnöki Kar Elektronikus Eszközök Tanszékén. A felajánlás keretében a Mentor több mint 20 millió dollár értékű elektronikai tervezőszoftverrel (EDA) járul hozzá a BME-n tanuló hallgatók mikroelektronikai-tervezési képzéséhez. A Mentor Graphics ehhez műszaki támogatást nyújt, oktatási anyagokat bocsát az egyetem rendelkezésére és ösztöndíjprogramot ajánl fel, amelyekre alapozva az egyetem korszerű analóg és vegyesjelű IC-tervező, oktató- és kutatóprogramokat indíthat el.

A Mentor Graphics Corporation a világ legsikeresebb elektronikai és félvezetőgyártó vállalatai számára kínál termékeket, valamint díjnyertes műszaki támogató és tanácsadói szolgáltatásokat. Az 1981-ben alakult cég az elmúlt esztendőben 725 millió dollárt meghaladó árbevételt realizált és több, mint 4000 embert foglalkoztat világszerte. A vállalatcsoport központja Oregonban, illetve a Szilíciumvölgyben található.

„Az új tervezőlaboratórium segíthet abban, hogy a fiatal magyar mikroelektronikai szakemberek felkészültebben vehessék fel a versenyt a nemzetközi versenytársakkal, és lehetővé tegyék, hogy egyre több IC tervezői munkahely létesüljön Magyarországon, így az itthon dolgozó tervezők nagyobb szellemet hasíthassanak ki az integráltáramkör-tervezés történetéből” – mondta Dr. Rencz Márta professzor, az Elektronikai Technológia Tanszék vezetője a szeptember 20-án megtartott ünnepélyes laborátadáson, amelyen a BME és a Mentor Graphics prominens vezetői is részt vettek.

