FIATAL MŰSZAKIAK TUDOMÁNYOS ÜLÉSSZAKA XIX.

Kolozsvár, 2014. március 20–21.

MONITOR KOMPONENSEK OKOSTELEFON PLATFORMRA

MONITOR COMPONENTS FOR SMARTPHONES

SZATMÁRI Zoltán⁽¹⁾, MICSKEI Zoltán⁽²⁾, MAJZIK István⁽³⁾

*Budapesti Műszaki és Gazdaságtudományi Egyetem, Villamosmérnöki és Informatikai Kar,
Méréstechnika és Információs Rendszerek Tanszék, H-1117, Magyarország, Budapest,
Magyar tudósok körútja 2; Telefon/Fax: +36-1-4634112, (1) szatmari@mit.bme.hu,
(2) micskeiz@mit.bme.hu, (3) majzik@mit.bme.hu*

Abstract

In case of smartphone platforms, there is an emerging need for monitoring the running applications based on the observable elementary events and sensor data, as well as for an active intervention in case of detecting specific composite events or event sequences. In our paper we describe the monitoring components that we developed for accessing elementary events and data, for the rule-based monitoring of the status of applications, and for the temporal logic based evaluation of event sequences. The novelty of our approach is the automated monitor synthesis based on the formal specification, and the handling of event taxonomy during the monitoring.

Keywords: smartphone, monitoring, monitor synthesis, event taxonomy

Összefoglalás

Okostelefon alkalmazások esetén is felmerül az igény a platformon megfigyelhető elemi adatok és események alapján a futó alkalmazások monitorozására, valamint a felismerhető összetett események illetve eseményszekvenciák esetén az aktív beavatkozásra. Cikkünkben ismertetjük azokat a monitor komponenseket, amiket az elemi adathozzáféréshez, a szabály alapú monitorozáshoz, valamint eseményszekvenciák temporális logikai specifikáció alapján történő kiértékeléséhez fejlesztettünk ki. Megoldásunk újdonsága a formális specifikáció alapján történő automatikus monitor szintézis, valamint az esemény taxonómia figyelembevétele a monitorozás során.

Kulcsszavak: okostelefon, monitorozás, monitori szintézis, esemény taxonómia

1. Bevezetés

Az okostelefonok fejlődésével és az egyre szélesebb körben történő elterjedésével igény jelentkezik a telefonokon elérhető és megfigyelhető információk alapján biztosított különböző szolgáltatásokra. A mobilkészülékeken számos elemi információ összegyűjthető, kezdve a különféle szenzor adatoktól (pl. pozíció és gyorsulás adatok) a hívási adatokon keresztül a felhasználói adatbevitelig. Ezen információk megfelelő megfigyelése, kategorizálása és kiértékelése lehetőséget nyújt olyan magasabb szintű szolgáltatások megvalósítására, amelyek az összegyűjtött adatokból és események sorozatából kikövetkeztethető összetett eseményekre nyújtanak programozott választ.

Ha egy egyszerű példaként elemi információnak tekintjük az idő és pozíció adatokat, valamint felhasználói bemenetként rendelkezésünkre áll egy egyetemi hallgató órarendje,

akkor a telefonon beállítható egy olyan automatizmus, ami egy előadás idejében, az előadóteremben automatikusan lehalkítja a készüléket. Ez az automatizmus szabályok segítségével specifikálható eseménymonitorozást és -kiértékelést igényel.

Egy másik példa lehet egy olyan elosztott adatgyűjtő alkalmazás, amelynek okostelefonra telepített komponense csak meghatározott protokoll szerint adhat át adatokat az adatelemző szerver komponensnek – ettől eltérő protokoll szerinti, adatbiztonság szempontjából kockázatos kommunikáció detektálása esetén a monitor leállítja az adattovábbítást és figyelmezteti a felhasználót. Ez utóbbi esetben ésszerű elvárás, hogy az elfogadható forgatókönyveket magas szintű modell (pl. üzenet-szekvencia diagramok, temporális logikai kifejezések) formájában specifikáljuk, és a lokális ellenőrzést végző monitort ebből a specifikációból származtassuk.

A cikk témája az adatok és események futásidőbeli elemzését és a releváns eseményszekvenciák felismerését megvalósító monitor komponensek megvalósítása, az elemi adathozzáférést és eseményleképezést végző komponensek mellett tárgyalva az ezekre épülő, szabályalapú kiértékelést és a forgatókönyv alapú temporális ellenőrzést megvalósító monitor komponenseket is. A javasolt megoldás újdonságai a következőkben foglalhatók össze:

- A szabályalapú kiértékelést végző monitor komponensek számára a szabályok formális, de intuitív nyelven fogalmazhatók meg, lehetővé téve akár az on-line szabálymegadást is.
- Az eseményszekvenciák felismerését végző monitor komponensek automatikus szintézissel állíthatók elő. Az elfogadható eseményszekvenciákat lineáris idejű temporális logikai (LTL) kifejezések segítségével specifikáljuk, így a monitor szintézis ezek alapján végezhető. Ugyanezek a kifejezések egy elosztott kommunikációs protokoll üzenet-szekvencia diagramja alapján is származtathatók, kiszűrve először az adott résztvevő monitor komponense számára releváns eseményszekvenciákat.
- A szabályok és eseményszekvenciák formális nyelven történő specifikálása miatt a monitor komponensek tulajdonképpen az alkalmazások futásidőbeli verifikációját végzik [1]. A meglévő monitor szintézis megoldásokhoz képest (pl. [2]) módszerünk újdonsága, hogy egy különösen hatékony futásidőbeli verifikációs algoritmust [3] alkalmaz, és ezt kibővíti az eseményhierarchia kezelésével. Ennek lényege, hogy a szabályokban illetve az LTL specifikációban magasabb hierarchiaszinten található események is megadhatók (amelyek segítségével egy elvárás általánosabb szinten fogalmazható meg), futásidőben ugyanakkor a monitor az ezek finomítását jelentő eseményeket is figyelembe veszi a kiértékeléskor. (Pl. egy általános „üzenetküldés” mint magas szintű esemény fennáll akár a „kérés küldés” akár a „nyugtaküldés” mint alacsonyabb szintű esemény bekövetkezésekor.) Az eseményfinomítási hierarchiát egy esemény taxonómia felvételével specifikálhatjuk.

A cikk további fejezetei az adathozzáférést és szabályalapú monitorozást (2. fejezet), a temporális logikai specifikáció alapján történő monitor szintézist (3. fejezet), valamint az esemény taxonómia figyelembevételét (4. fejezet) ismertetik.

2. Elemi adathozzáférés és szabályalapú monitorozás

Az elterjedt mobilplatformok számos beépített lehetőséget biztosítanak, hogy információkat szerezzük a telefon hardveres és szoftveres állapotáról, a rajta futó alkalmazásokról vagy akár a telefon szenzoradatain keresztül közvetlen a környezetről. Az 1. táblázat egy konkrét platform, az Android példáján keresztül szemlélteti, hogy milyen tipikus jellemzőket lehet lekérdezni. Ezen jellemzők lekérdezését természetesen minden platform

megfelelő jogosultságok meglétéhez köti, de ha a monitorozó alkalmazás ezekkel rendelkezik, akkor sokféle helyzetet képes ezek alapján felismerni.

1. táblázat. Az Android platformon megfigyelhető elemi jellemzők (részlet)

Típus	Tipikus jellemzők
OS adatok	CPU-terhelés, memórafoglaltság, futó folyamatok száma
Szenzor adatok	Helyzet (pl. GPS), fényerő, gyorsulás
Platform adatok	Hálózati forgalom, alkalmazások állapota

Egyszerűbb szituációk és forgatókönyvek megfogalmazhatóak egy szabályalapú kiértékelő segítségével. Az általunk használt monitorozó komponens *HA (feltétel) AKKOR (beavatkozások)* típusú szabályok kiértékelését végzi el, ahol a feltétel egy logikai kifejezés, amiben az elemi jellemzőkre megfogalmazott konstansokat, egyenlőséget és relációs operátorokat tartalmazó összehasonlítások szerepelnek. A beavatkozások pedig végrehajtható akciók sorozata, pl. adott alkalmazás leállítása vagy hálózati forgalom letiltása. Ennek a segítségével már a gyakorlatban használható szabályok is felírhatóak, például:

```
HA ((akku_töltöttség < 20%) ÉS (alkalmazás_cpu_használat > 80%))
AKKOR (alkalmazás_leállítás)
```

Ha ennél bonyolultabb időbeli feltételeket szeretnénk megfogalmazni, akkor használhatóak a temporális logikai kifejezések alapján generált monitor komponensek.

3. Monitor komponensek szintézise temporális logikai specifikáció alapján

A temporális logikai specifikációt akár közvetlenül felhasználhatjuk a logikai időbeli (azaz események, állapotok egymásutánosságát megfogalmazó) követelmények megfogalmazására, akár pedig köztes reprezentációként származtathatjuk elosztott alkalmazások forgatókönyveinek egy-egy résztvevőre vonatkozó követelményeinek kiszűrése során. Munkánk során az LTL variánst használtuk fel, amelynek a Boole logikai operátorokat (And, Or, Not) kibővítő temporális operátorai az X (következő állapotban), F (valamikor a jövőben), G (folyamatosan) és U (addig, amíg). Ezek az operátorok egyszerű eseményekre (pl. üzenetküldésre, elemi adatok vagy állapotok változására) vonatkoznak.

A monitor szintézis alapötlete, hogy az LTL kifejezést ekvivalens átalakításokkal a következő formába hozzuk: először az F és G operátorokat az X és U segítségével fejezzük ki, majd az U operátorokat X operátorok hatókörébe helyezzük át [3]. Ez alapján a kifejezés kiértékelését végző monitor komponens alábbi blokkjai (függvényei) különböztethetők meg:

- Egy Boole logikai operátorokat megvalósító blokk, ami az adott állapotban megfigyelt eseményeket és azokat a blokkokat köti össze, amik az X operátorok utáni részkiefejezéseket értékelik majd ki. Mivel ez utóbbiak kiértékelésének eredménye az adott állapotban még nem ismert, azokat háromállapotú logikával, „ismeretlen” értéként vesszük figyelembe (az „ismeretlen” értékek a következő blokkok eredménye alapján helyettesíthetők majd be). Ezen blokk forráskódját tehát a háromértékű logikán végzett műveletek határozzák meg. Amint ez a kiértékelés konkrét (igaz vagy hamis) eredményt ad, az a monitor kimenetén megjelenik.
- Egy-egy, X operátor utáni részkiefejezést megvalósító blokk, amelynek kiértékelése a futás következő állapotában (a megfigyelt jellemzők változása után) következhet. A blokk felépítése újra a fentiek szerint történik, azaz a részkiefejezés ekvivalens átalakításával, majd az adott állapotban lokálisan kiértékelhető blokk és a következő állapotra vonatkozó további blokkok megkülönböztetésével.

Mivel az eredeti kifejezés véges számú operátort tartalmaz, a különböző felépítésű blokkok száma is véges lesz [3]. Az ezekhez tartozó forráskód részleteket (függvényeket) a monitor kódjának generálása során hozzuk létre, így futásidőben csak ezek hívását, a kiértékelés eredményének nyilvántartását és az események alapján történő folyamatos frissítését kell elvégezni. A kódgenerálás során szükség volt az LTL kifejezések reprezentálására és átalakítására, valamint a blokkok kódjának optimális (gyorsan kiértékelhető és kis memóriaigényű) megvalósítására.

4. Esemény taxonómia figyelembevétele a monitorozás során

Az előzőekben bemutatott szabályok és temporális logikai kifejezések elemi eseményekből, mint alkotóelemekből épülnek fel. A kifejezések általánosabb leírhatóságára vonatkozó igény motiválja az eseménytípusok kategorizálását és ez alapján egy esemény taxonómia felépítését. Ennek megfelelően az okostelefon környezetben részleteztük és összegyűjtöttük a korábban feltérképezett eseménytípusokat és elkészítettük az ezek alapján felállítható taxonómiát. Így például definiáltuk a „mozgás” eseményt, ami akár a gyorsulásmérőből származó esemény, akár pedig a cellainformáció, GPS koordináták, vagy az elérhető vezeték nélküli internetkapcsolatok halmazának megváltozása események általánosított öse lett. A taxonómia építés során több esetben felmerült, hogy egy elemi esemény több általánosabb esemény alá is tartozhat, így engedélyeztük a többszörös öröklés lehetőségét.

Az így elkészített taxonómia figyelembevételét (hasonlóan a [4]-beli megoldáshoz) beépítettük a monitorozó komponenseinkbe, és az implementáció kis arányú módosításával kiterjesztettük a leírható kifejezések és ezáltal a futásidőben ellenőrizhető szituációk körét.

5. Összefoglalás

Munkánk során megvizsgáltuk, hogy az elterjedt okostelefon platformokon milyen elemi információk figyelhetőek meg, majd ezekhez illeszkedve megoldásokat adtunk szabály alapú monitorozásra illetve temporális logikával megadott eseményszekvenciák kiértékelésére, végül pedig kidolgoztuk, hogy az események taxonómia alapú kategorizálását hogyan lehet figyelembe venni a monitorozás során. Az adathozzáféréshez könyvtári függvényeket használó, a magas szintű specifikációk alapján pedig automatikus szintézissel előállított monitor komponensekből mint modulokból egy olyan köztesréteg jellegű szolgáltatás állítható össze, ami alkalmazások széles köre számára biztosíthatja a futásidőbeli verifikációt illetve a felismert eseményszekvenciák és összetett események programozott kezelését.

A kutatást a BME VIKING Nonprofit Zrt részvételével az „Infokommunikációs technológiák és a jövő társadalma FuturICT.hu” című, TÁMOP-4.2.2.C-11/1/KONV-2012-0013 azonosító számú kutatás-fejlesztési projekt támogatta.

Irodalom

- [1] Sokolsky, O., Havelund, K., Lee, I.: *Introduction to the Special Section on Runtime Verification*. Int J Software Tools Technology Transfer, 14(3), pp 243-247, Springer Verlag, 2012.
- [2] Finkbeiner, B., Sipma, H.: *Checking Finite Traces using Alternating Automata*. In Electronic Notes on Theoretical Computer Science, Vol. 55, Elsevier, 2001.
- [3] Pintér, G., Majzik, I.: *Automatic Generation of Executable Assertions for Runtime Checking Temporal Requirements*. In: Proc. HASE 2005, pp 111-120, IEEE CS, 2005.
- [4] Horányi, G., Micskei, Z., Majzik, I.: *Scenario-based Automated Evaluation of Test Traces of Autonomous Systems*. In Proc. of ERCIM/EWICS Workshop on Dependable Embedded and Cyber-physical Systems, Toulouse, France, 2013.