



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Automatizálási és Alkalmazott Informatikai Tanszék

Zöldi-Kovács Róbert

Kristóf Timur

WORKFLOW KEZELŐ
ALKALMAZÁS
WEBES KISZOLGÁLÓJA

KONZULENS

Dr. Forstner Bertalan

BUDAPEST, 2012

Tartalomjegyzék

Tartalomjegyzék.....	2
1. Bevezetés	4
2. A munkafolyamat kezelés hatékonysága	5
3. Az MFlow rendszer koncepciója	6
4. Felhasználhatósági példák	11
4.1. Biztonsági őr.....	11
4.2. Futárszolgálat.....	11
4.3. Utazó ügynök.....	11
4.4. Külső helyszínekre érkező kérések feldolgozása	12
5. Tervezés, prototípus megvalósítása	12
5.1. A rendszer architektúrája.....	12
5.2. Felhasznált technológia	13
5.3. A prototípus funkcionalitása.....	14
5.4. A prototípus belső működése	16
6. Az MFlow munkafolyamat szerkesztő	17
6.1. Elvek és motiváció.....	18
6.1.1. Háttér.....	18
6.1.2. Célunk	18
6.1.3. Építőelemek.....	18
6.2. Felhasználói felület leírása	19
6.3. Felhasznált technológia	22
6.3.1. Háttér.....	22
6.3.2. Racionalitás	22
6.3.3. Előnyök	22
6.3.4. Felhasznált megoldások (kliens oldal).....	23
6.3.5. Felhasznált megoldás (szerver oldal).....	23
6.4. Működés	24
7. Mobil kliens minta	25
8. Használati példák a rendszerben	27
8.1. Biztonsági őr.....	27
8.2. Futárszolgálat.....	28
9. Összefoglaló.....	28

9.1.	A jelenlegi állapot.....	28
9.2.	Továbbfejlesztési lehetőségek	29
10.	Forrásgyűjtemény.....	29

1. Bevezetés

Számos vállalkozásnál lehet látni, hogy ahogy túlnő egy bizonyos határon a munka mennyisége, és az alkalmazottak száma, a menedzsment egyre nagyobb problémákba ütközik, megfelelő eszközök híján a munka kiosztása, felügyelete, és végrehajtása során egyre több hiba halmozódik fel. Ez mind a munkaerőre, mind az ügyfelekre rossz benyomást tesz. Ezen problémák orvoslására a cégek változatos menedzsment rendszerek bevezetése mellett döntenek, ám nagyon gyakran ezek nem segítenek, vagy csak súlyosbítanak a problémákon. Az ilyen rendszerek bevezetése eleve nehéz, költséges feladat, még nyílt forrású rendszerek esetén is, a használatra való képzés is drága, és időigényes feladat. Ezek mellett a legtöbbször ezek a rendszerek a nagyvállalatok igényeit helyezik előtérbe, a kis- és középvállalatok számára túl sok, bonyolult szolgáltatást nyújtanak, és legtöbbször a munkafolyamatok kezelését nem, vagy csak korlátozottan támogatják.^[1] Számos szoftver létezik, amely projektmenedzsmentre, adatok, dokumentumok megosztására, és a feladatok task szintű kiosztására specializálódnak, de a feladatok végrehajtásának folyamatát már nem tartalmazzák. Más szoftverekbe üzleti folyamatokat be lehet vinni, és optimalizálni, de a feladatkiosztást nem támogatják. A legnagyobb szoftverek közül például a Microsoft Dynamics CRM nevű szoftver (amellett, hogy már a nevéből is látszik, hogy nem az üzleti folyamatok kezelése a célja) képes kezelni a szerződéseket SLA szinten, az ügyfeleket kapcsolattartókhöz rendelni, taskok kiosztását és végrehajtását felügyelni, és még rengeteg más szolgáltatást nyújt, amire csak egy nagyvállalatnak van szüksége, valamint az ára is ehhez viszonyul, de az üzleti folyamatok menedzsmentjét (BPM) csak egyedi fejlesztések segítségével képes támogatni, amelyet jól megcsinálni igen költséges, hosszú folyamat.^[2] Az üzleti folyamatok futtatására például a Microsoft BizTalk Server^[3], vagy a Process Maker^[4] névre hallgató open source alkalmazás alkalmas, de egyik sem használja ki a mobil kliensek adottságait. Emellett a Microsoft terméke csak a nagyvállalatok számára kifizethető árban mozog.

Nagyon sok helyen előfordulhat olyan eset, amikor a munka elvégzése közben nem elérhető megfelelő számítógép az alkalmazottak számára. Ez körülményessé teszi az elvégzett munka naplózását, az új feladatok kiosztását, a többszereplős folyamatok folyamatos végrehajtását. Jogos elvárás tehát, hogy biztosított legyen egy mobilos felület az alkalmazáshoz, amely elérhető a legtöbb mobil platformon, legalább mobil böngészőre optimalizált weboldal segítségével. A megnézett szoftverek szolgáltatásai között csak ritkán szerepelt a mobil eszközökre optimalizált felület, és egyik sem használta ki az ilyen eszközök szolgáltatásaiban rejlő potenciált.

Észrevettük, hogy a munkafolyamat-kezelés ezen két problémája egyáltalán nem zárja ki egymást. Az egyszerűség, amit a mobilos felület korlátozott mérete megkövetel kívánatos a kisebb vállalkozások számára, míg az eszközök által felkínált szolgáltatások támogatása a munkafolyamatokban teljesen új megközelítést ad a piacon. A munkafolyamatok egyszerű kezelésére, és a mobil eszközök kihasználására irányuló elvárásokat ötvözve megszületett az MFlow névre keresztelt mobil munkafolyamat kezelő alkalmazás ötlete, amely mobil

kliensekre optimalizált szolgáltatás nyújtását tűzi ki célul, a kis- és középvállalkozások igényeit középpontba helyezve.

Úgy véljük, hogy a vállalatoknak törekedni kell arra, hogy elsősre a megfelelő eszközt válasszák a problémáik megoldására, és ez a koncepció egy olyan új eszközt vázol fel, amely betölthet egy egyedi helyet a BPM rendszerek között, több olyan szolgáltatást egyesítve, és középpontba helyezve, amelyek egyszerre nem szerepelnek egy rendszerben sem, és kihasználja a mobil kliensek által nyújtott lehetőségeket, miközben mellőzi a feleslegesen bonyolult szolgáltatásokat.

Jelen dolgozat a koncepció kidolgozását, valamint egy prototípus alkalmazás tervezési folyamatát, és implementációját tárgyalja, használati példákkal is szemléltetve a célt.

2. A munkafolyamat kezelés hatékonysága

Az elvégzett munka minőségét, és a folyamatok hatékonyságát több módszerrel is vizsgálhatjuk. Ilyen például az ISO-9001 szabványnak való megfelelés, amelyhez komoly feltételeket kell teljesíteni a minőségirányítás terén. A vállalatnak demonstrálnia kell a minőségirányítási rendszer meglétét, minőségpolitikáját, és minőségcéljait, valamint ezeknek a betartását is.^[5]

Ugyan már elavult módszer, de régebben a CMM (Capability Maturity Model) használatával értékelték az informatikai cégek működését.^[6] Ennek a különböző szintjeihez hasonlóan, általánosítva leírhatjuk egy vállalat folyamatainak fejlődését.

A legegyszerűbb munkafolyamat-kezelés során nincsen semmilyen előre definiált folyamat, protokoll. A feladat kiadása után mindenki úgy csinálja, ahogy akarja. Az egymással való kommunikáció esetleges, a módja véletlenszerű. A folyamatok megismételhetetlenek, és utólag gyakorlatilag lehetetlen kideríteni, hogy mi is történt, ha valami rosszul alakul, felelőst találni csak az alkalmazottak őszinteségére alapozva lehet. Az, hogy egy munka egyáltalán elkészül-e az emberi emlékezeten múlik.

Ezen a kezdeti, esetleges szinten minden vállalkozásnak túl kell lépni a működés legelején, hiszen ahogy több munka érkezik be, szinte lehetetlenné válik minden ügyfél adatait, dokumentumait fejben tartani, és előkeresni valahonnan a saját számítógépekről.

Az elvégzendő, illetve elvégzett munkát különféle dokumentumokban tárolják, dátummal, illetve az azt elvégző alkalmazott nevével együtt. A szerződések, dokumentumok is egy közös helyen tároltak, bár ennél több rendszerezés nem jellemző. A munkafolyamat még mindig nem, vagy csak nehezen követhető, de már a kezdemények megvannak. A dokumentumok frissen tartása emberi feladat, kimaradhat bejegyzés, frissítés, vagy éppen többször, több helyre kerülhetnek be egybetartozó bejegyzések. A helyes formát is az embereknek kell fejben tartani, és betartani. Felelőst találni a munkákhoz már lehetséges, de a kommunikáció még mindig követhetetlen, a folyamatok megismételhetetlenek. A rendszer működése még

mindig az emberek korrektségén múlik, hiszen gyakorlatilag követhetetlen, hogy ki és mit írt bele egy dokumentumba.

Ez a módszer már működhet a legkisebb vállalatok szintjén, ahol mindenki ismer mindenkit. Ám ahogy elkezdi nőni a vállalat, kialakulnak ellentétek, a vezetőség jobban elkülönül az alkalmazottaktól, és elkezdődnek a problémák. Előbb-utóbb valaki megpróbálja kihasználni a rendszer úgymond naivitását, és ez hamar a produktivitás kárára megy. Ezt elkerülendő, a menedzsment egyre inkább át kell, hogy térjen a „tűzoltásról” a minőség fejlesztésére. Ekkor bevezetésre kerülnek dokumentum kezelő rendszerek, az ügyféllel való érintkezés többé-kevésbé megszervezésre kerül, a munka kiosztása, és számonkérése egyre pontosabban, szigorúbban történik.

A munka mennyiségének növekedése viszont az emberi hibák lehetőségét még mindig megnöveli. Ekkor jön a képbe a munkafolyamat kezelés, amely segítségével szabványos folyamatokkal lehet leírni a vállalat profilját, a munka jelentős része ezen munkafolyamatok végrehajtásából áll. Az MFlow rendszert az erre a szintre törekvő vállalatokat szem előtt tartva tervezzük.

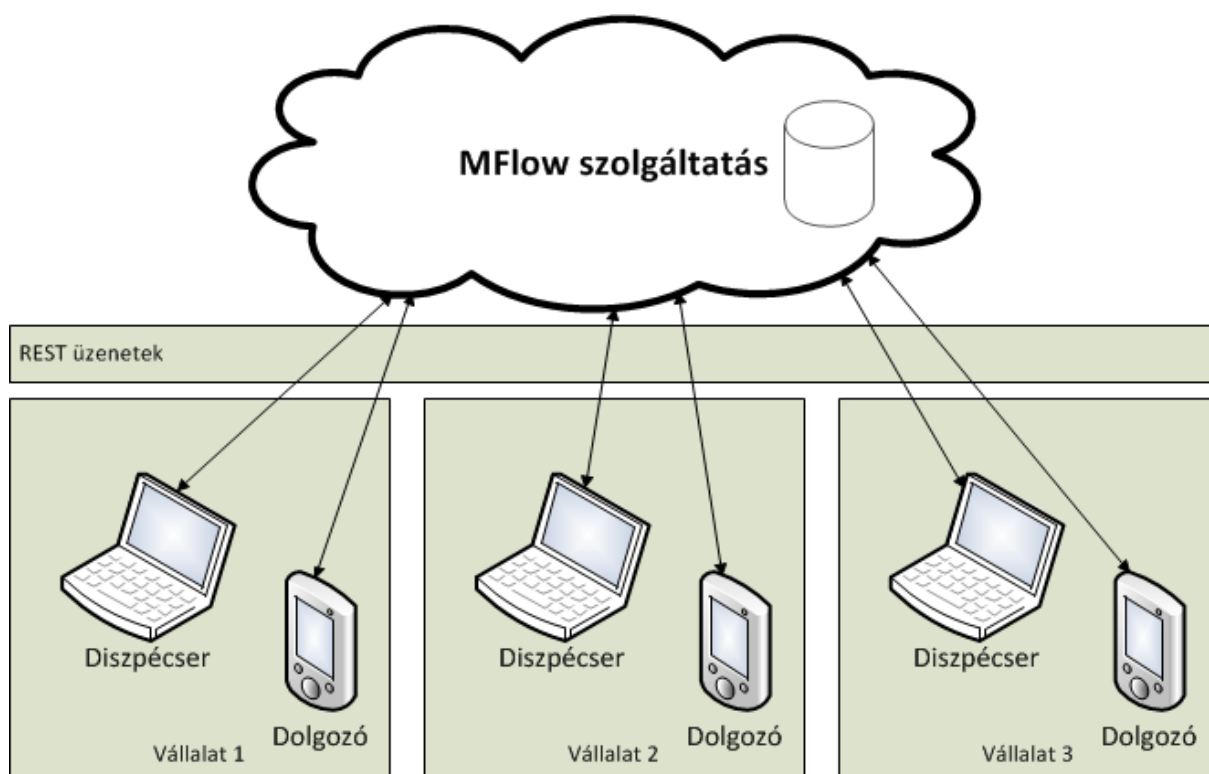
Természetesen a fenti szintek a valóságban nem különíthetők el ilyen élesen, egyetlen vállalat sem pontosan így fejlődik, nem ezek a problémák jelentkeznek, de az általános folyamatot jól jellemzik.

A munkafolyamat kezelő rendszerek bevezetésének hatékonyságát számos cikk, és ipari példa igazolja. Minden ilyen forrás egyetért abban, hogy a folyamatok helyes menedzselése nagyban csökkenti azok végrehajtási idejét, és ezáltal is a költségét.^[7] Az AAIT tanszék projektjei között is található erre példát, a rendszer egy korábbi prototípusának bevezetése a Nokianál a PIP (Project Incentive Plan) folyamatot, vagyis a bónuszkifizetések meghatározási és jóváhagyási folyamatát 12 hétről 4 hétre csökkentette, ami nagyon jelentős javulás.

3. Az MFlow rendszer koncepciója

A koncepció főbb pontjai:

- Munkafolyamat kezelést valósítson meg a rendszer
- Mobil kliensekre optimalizált felület kialakítását tegye lehetővé
- Mellőzze a felesleges szolgáltatásokat
- Szolgáltatás alapon működjön (egy kiszolgáló számos ügyfelet szolgáljon ki)
- A felhasználókkal üzenetekkel, illetve válaszüzenetekkel kommunikáljon, ne igényeljen folyamatos kapcsolatot.



1. ábra Szolgáltatás koncepció

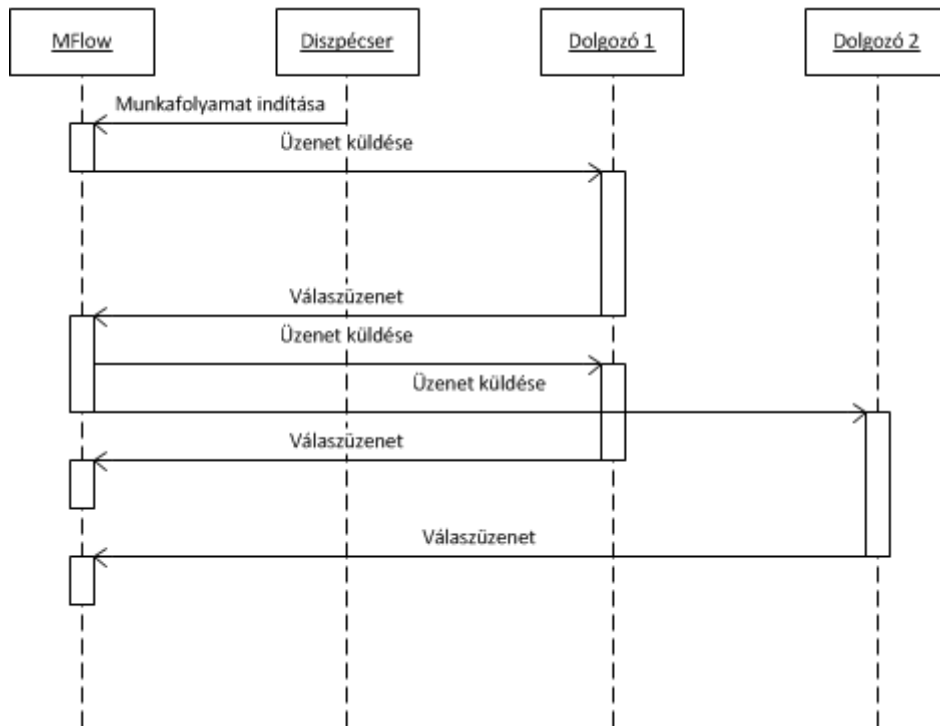
Az 1. ábra szemlélteti a szolgáltatás koncepcióját. Az ügyfelek elkülönülnek egymástól, nem férhetnek egymás adataihoz, egy felhasználó csak egy vállalathoz tartozik a rendszerben (még ha több ügyfélnél is szerepel egy adott természetes személy, a rendszer akkor is külön felhasználóként fogja kezelni), viszont a kiszolgáló oldalán az adatokat egy szolgáltatás kezeli, és egy adatbázisba kerülnek.

Az elképzelés szerint a felhasználóknak nem kellene a licenszért fizetni, csak valamilyen alapsomag, amely néhány felhasználót tartalmazna, illetve a további felhasználók után, megfelelő árazás szerint havidíjban fizetni. Ez különösen vonzó lehet a kis- és középvállalkozások számára, akik nem, vagy csak nehezen engedhetik meg maguknak egy drága bonyolult rendszer megvásárlását. Példának okáért a Microsoft BizTalk Server licenszdíja 10000 dollárnál kezdődik^[8], és még fenn is kell tartani. A rendszer használatára oktatóvideók állnának rendelkezésre, illetve külön díjért akár oktatást is lehetne rendelni a felhasználók számára.

Az egyedi megoldásokat az architektúrának köszönhetően kevésbé tartalmazná a rendszer, de például egyedi adminisztrációs felület fejlesztése, vagy saját honlapba beágyazása könnyen kivitelezhető, hiszen ezek csak a megjelenést változtatják meg. Az ennél egzotikusabb kérések egyéni elbírálást igényelnének, de valószínűleg a legtöbb esetben valószínűleg nem lennének teljesíthetőek, csak ha elég nagy igény jelentkezne, hogy megérje a rendszeren nagyobb méretű fejlesztést végrehajtani.

A munkafolyamatok megtervezéséért, és elindításáért egy asztali böngészős webalkalmazás felel, amelyet a diszpécser, vagy adminisztrátor kezel. Ez a felület lesz alkalmas a felhasználók kezelésére, előfizetési információk lekérésére, és számos egyéb adminisztratív

funkció végrehajtására is. A mobil alkalmazást használó dolgozók ezen alkalmazás segítségével lekérlik az üzeneteiket a szervertől, végrehajtják a bennük található utasításokat, majd kitöltik és visszaküldik a szervernek az űrlapot, ezáltal naplózva az elvégzett munkát.



2. ábra Munkafolyamatok végrehajtása koncepció

A 2. ábrán látható diagram egy munkafolyamat lefutását vázolja. Látható, hogy minden kommunikáció a szerveren keresztül történik, ezáltal biztosítva, hogy amennyiben csak a rendszer által biztosított munkafolyamatok keretén belül történik a munka, nem veszik el információ, minden dokumentálva megtalálható. Mivel a munkafolyamat léptetését, és az üzenetek elküldését a rendszer automatikusan végzi, a feladatok kiosztásának folyamata jelentősen gyorsabbá válik.

Fontos a munkafolyamat értelmezése. Esetünkben ez alatt egy vagy több ember által elvégzendő munkát értünk, lebontva a lényegi döntési, illetve adatot generáló pontokra. Az elképzelés szerint ezek végrehajtására időkorlát is adható, illetve időzített események köthetők hozzá. Például ha egy adott ponton valakinek el kell vállalnia egy feladatot, akkor a rendszer véletlenszerűen, vagy valamilyen algoritmus szerint kiválaszthat valakit, akinek továbbítja, vagy egy feletttest megkereshet, hogy adja valakinek a feladatot.

A rendszer működésének egy üzenet alapú megoldást találtuk a legjobbnak. A munkafolyamatok minden elért pontján a kiszolgáló üzenetet küld a folyamat aktuális lépéséhez rendelt felhasználói csoport tagjainak, amely lényegében egy űrlapot tartalmaz. A mobilos kliens egy szolgáltatáshívással lekéri az üzeneteket, majd a kitöltés után hasonló módon válaszüzenetet küld. A beérkezett válaszok, és egyéb események alapján a rendszer eldönti, hogy tovább lehet-e léptetni a folyamatot, és ha igen, milyen irányba. A koncepció szerint lényegében tetszőleges bonyolultságú munkafolyamat megadható, amely önmagán

belül dinamikusan létrehozhat munkacsoportokat a válaszok alapján, működhet ciklusban, elágazhat, vagy lezárhatja magát.

A mobil kliensekre optimalizált felület egyszerűséget követel meg, nem lehet sokmindent megjeleníteni egy oldalon, illetve az egész alkalmazásnak átláthatónak kell lennie. Ennek köszönhetően fölösleges túl bonyolult adatokkal dolgozni, túl sok szolgáltatást belezsúfolni a rendszerbe. Szigorúan a munkafolyamat kezelés a cél, valamint a köré épülő szolgáltatás biztosítása. A rendszer által küldött üzeneteknek is egyszerűen érthetőnek, megjeleníthetőnek kell, hogy legyenek.

Ugyanakkor a mobil kliensek megjelenítési képességeit ellensúlyozza a mobilitásuk, és számos szolgáltatás amit nyújtanak. Mára a legtöbb okostelefonban található GPS, kamera, és rengeteg egyéb szenzor, amelynek az adatait az űrlap kitöltése során le lehet kérni, és visszaküldeni a szervernek. Az emberek munkafolyamatok lépéseikhez való hozzárendelésére is alkalmazhatóak ezek a módszerek, például lehetséges, hogy a rendszer automatikusan a célhoz legközelebb eső ember jelöli ki a feladatra, a kliens által biztosított GPS koordináta alapján. A kamera megléte lehetővé teszi, hogy az űrlap kitöltése közben a dolgozó képet készítsen, és azt is visszaküldje a válaszüzenetben. Asztali környezetben ennek a kivitelezése körülményes feladat.

A 3-4. ábrán látható, hogy az üzenetek megjelenítésére az Androidos klienst készítő milyen felületet terveztek a munkafolyamat végrehajtása során küldött űrlapok megjelenítésére. Megfigyelhető, hogy a sorrendben megjelenő űrlap elemek mind érintőképernyőn kényelmesen kezelhető méretűek, az egész űrlap könnyen átlátható, és nem kell például oldalra scrollozni az olvasáshoz. A bonyolultabb vezérlők, mint például egy térképkoordináta, vagy dátum doboz beviteli képernyőt adnak a felhasználó elé, amelyen könnyebb bevinni az adatokat. Ezek a megoldások minden klientípuson eltérőek lehetnek, az operációs rendszer stíluát használva a kényelmes használat érdekében. Windows Phone operációs rendszeren például a legördülő lista típusú elemek is egy választóablakot adnak, míg androidon tényleg csak egy legördülő lista jelenik meg^[9].

Jelölőnégyzet felirat

Szövegdoboz cím

Szövegdoboz

Rádiógomb csoport cím

Opció #1

Opció #2

Opció #3

Legördülő lista cím

Opciók listája ▼

3. ábra Az Androidos kliens felület-terve

Szövegdoboz cím

Aktuális érték

Térképelem cím

Térképre!

Naptárelem cím

Hozzáadás naptárhoz

Fájlcsatolmány cím

Letöltés Megnyitás

Dátum cím

Szövegdoboz

Tuesday, April 12, 1977

Apr 12 1977

Set Cancel

4. ábra Az Androidos kliens felület-terve

4. Felhasználhatósági példák

4.1. Biztonsági őr

Az MFlow rendszer előnye személtethető például egy biztonsági őrököt biztosító vállalat példájával. Léteznek kiépíthető rendszerek, amelyek segítségével követhető az őr mozgása, de erre nem mindenhol van lehetőség, valamint a költségeket figyelembe véve olcsóbb lehet az MFlow munkafolyamatokat használó megoldás. A legszemléletesebb egy nagy területet lefedő őrjárat munkafolyamata. A folyamat során a különböző bejárando pontok egy-egy üzenetben kerülnek meghatározásra, amelynek elérésekor a válaszüzenetben az automatikusan kitöltött GPS-koordináta mellett visszakérülhet, hogy történt-e valamilyen incidens. Incidens esetén egy elágazás történik a folyamatban, amelyben az incidens részleteiről lehet jelentést tenni. A végeredmény, hogy az ügyfél számára bizonyítható az előírt szolgáltatás biztosítása, még ha nincs is kiépített rendszer az őrök követésére. Esetleg egy nagyobb, automatizált munkafolyamat az őrjáratok időben történő indításáról is gondoskodhat, tovább csökkentve az emberi hiba lehetőségét.

4.2. Futárszolgálat

Az MFlow rendszert egy futár szolgáltatást nyújtó vállalkozás kiválóan tudná alkalmazni. A központban lévő diszpécserhez beérkező megrendelések alapján indít munkafolyamatokat. A kiindulási, és végpont alapján különböző futarcsoportoknak elküldheti a feladatot, ahol az első elérhető ezt elvállalja, és elmegy a csomagért. A csomag felvételekor jelez a rendszernek, elküldve az aktuális GPS koordinátát, és egy fényképet a csomagról, majd a csomag megérkezésekor hasonlóan jelzi annak megérkezését, és sikeres kézbesítését. Amennyiben a kézbesítés sikertelen, értesíti az ügyfelet, és a válasz alapján visszaviszi, vagy megőrzésre elhelyezi egy központi helyen. Innen egy - akár másik - futár később ismét megpróbálja a kiszállítást. Ez a folyamat a diszpécser által akár előre, akár menet közben meghatározott ideig folytatódhat. A szolgáltatás díjának kifizetését eközben a munkafolyamat egy másik ága jelzi, a válaszott fizetési módtól függően, akár előre fizetést is figyelembe véve (bár az ügyfél számára elérhető keret ellenőrzését a diszpécsernek kell megtenni, erre már nem terjed ki a szolgáltatás.)

4.3. Utazó ügynök

Az MFlow rendszer kiválóan alkalmazható egy utazó ügynököket alkalmazó vállalat esetén. Akár felmérést végez, akár villanyórát olvas le, akár megpróbál valamit eladni, minden meglátogatott célpont leírható egy munkafolyamat szakasz segítségével, amely ciklusban ismétli önmagát. Ezáltal követhetővé válik az ügynök munkája, hogy tényleg meglátogatta-e a célokat, hol engedték be, szükség esetén később újra visszatért-e, vagy, hogy mi volt az eredménye a látogatásnak. Valamilyen termék eladása esetén akár a megrendelés folyamata is a munkafolyamat részét képezheti, vagy egy másik munkafolyamat elindítását eredményezheti.

4.4. Külső helyszínekre érkező kérések feldolgozása

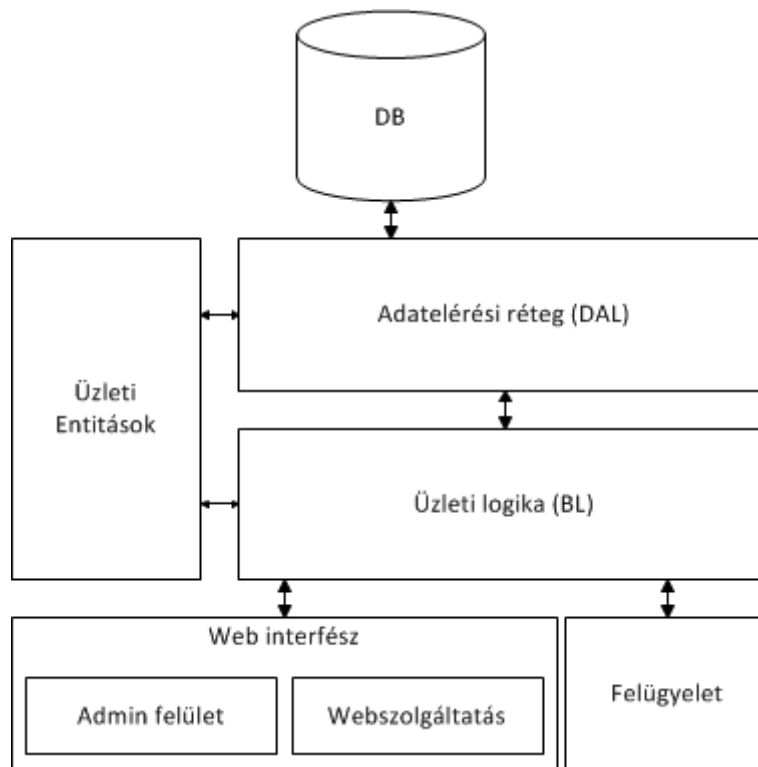
Egy sok telephellyel rendelkező üzlet esetén nagyban gyorsíthat az ügyintézésen egy munkafolyamat-kezelő alkalmazás. Például ha nem egyértelmű egy esetben a garanciális feltételek teljesülése, telefonálgatás, a termék szállítgatása helyett néhány kép elkészítésével, és elküldésével továbbítani lehet a kérdést eldönteni hivatott ember felé a problémát, aki azonnal értesítést kap, és dönteni tud. Más esetben, ha például helyben nem tudják a terméket javítani, vagy cserélni, a problémamegoldás folyamata során fellépő kommunikáció nagyban csökkenthető, és nyomomonkövethető egy ilyen munkafolyamat-kezelő alkalmazás segítségével.

5. Tervezés, prototípus megvalósítása

A koncepció működésének demonstrálására egy prototípus alkalmazás elkészítését tűztük ki célul. Korlátozott mennyiségű, de a végcélt szemléltető szolgáltatás implementációját terveztük meg, elsősorban a munkafolyamat-kezelés témakörében, a végső terveknel egyszerűbb, de bővíthető lehetőségekkel, míg például az autorizációnak csak az alapjait terveztük beletenni, tényleges működés nélkül. Ez utóbbira például egyáltalán nincs szükség a koncepció helyességének megmutatásához.

5.1. A rendszer architektúrája

A prototípus rendszer megvalósításához háromrétegű architektúrát alkalmaztunk, elválasztva az adatbázist, az üzleti logikát, és a webes interfészt, amely a kliensek felé továbbítja a megjelenítendő adatot, illetve fogadja a válaszokat, és lekéréseket. A webes interfész két részből áll, a böngészőből elérhető adminisztrációs felületből, illetve a mobilos kliensek kiszolgálására kialakított JSON adatokat továbbító szolgáltatásból. A publikus interfész mellett egy felügyeleti interfészt is terveztünk, amely a rendszer felügyeletét, és karbantartását teszi lehetővé, de a jelenlegi implementáció ezzel nem foglalkozik. Az 5. ábrán a rendszer architektúrájáról láthatunk egy vázlatot.



5. ábra Az MFlow szolgáltatás architektúrája

5.2. Felhasznált technológia

A prototípus elkészítéséhez a C# nyelvet választottuk, és a szolgáltatás futtatását Windows Azure Cloud Services környezetben terveztük futtatni. A prototípus futtatásához egy Web role, és egy adatbázis elegendő, egy későbbi implementáció esetén egy Worker role bevezetése megfontolandó.

Az adatbázist code first eljárással terveztük meg, az entitások elkészítése közben megalkottuk a kapcsolatokat, illetve relációsan nehézkesen tárolható adatok esetében szerializált tárolást elrendelve. Az adatelérési réteg tervezése során a Fluent Nhibernate^[10] technológiát választottuk, amelyet konfigurálva a megfelelően elkészített osztályokból képes relációs adatbázist generálni. Az adatbázis eléréséhez egy Repository osztályt használunk, amely később viszonylag könnyen frissíthetővé, vagy akár lecserélhetővé teszi az adateléréshez használt technológiát, ha szükséges. Az NHibernate technológia működik a Windows Azure^[11] relációs adatbázisával is, bár egyelőre az adatbázist csak hagyományos SQL Server segítségével teszteltük le.

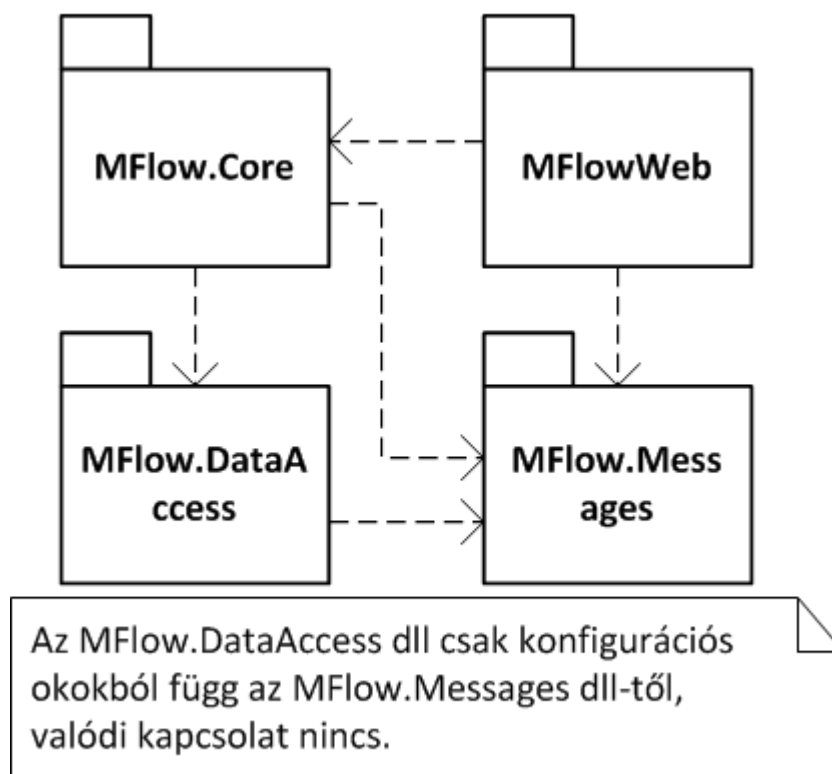
A szerver által küldött üzeneteket reprezentáló objektumok egy Portable Library Tools^[12] segítségével készült assemblybe terveztük meg, amelyre referenciát tud adni például a Windows Phone kliens is, ezáltal könnyítve a fejlesztést.

A tervezés során világossá vált, hogy bizonyos funkciók kényelmes, biztonságos megvalósításához bizonyos entitások interfészéhez az alkalmazás magja mellett a webes interfésznek is hozzá kell férnie. Mivel ezek az entitások nem tartalmazznak tényleges üzleti

logikát, és az interfészük önmaguk, ezért az üzeneteket tároló assemblybe helyeztük őket, és a magból referenciát adtunk rá. Ez a döntés némiképp megbontja a rétegek szeparációját a fájlok szintjén, de a kód szintjén nincs érintkezés az üzleti logika, és a webes felület között az egymás felé nyújtott interfészen kívül, valamint az üzenetek assemblyje nem tartalmaz referenciát a magra, tehát továbbra sincs szükség a tényleges üzleti logikára referenciát adni a kliensek fejlesztése során.

A Webes interfész kialakításához ASP.NET MVC 4^[13] technológia használata mellett döntöttünk, Razor View Engine használatával. Az ASP.NET MVC 4 a JSON adatok sorosítására a JSON.NET assemblyt használja fel^[14].

A 6. ábrán látható a különböző MFlow Assembly egymáshoz való viszonya. Az MFlow.DataAccess, és az MFlow.Messages assembly közötti kapcsolat csak az adatelérési réteg konfigurációja miatt van jelen, és egy későbbi verzióban megszüntetjük.



6. ábra Az MFlow assemblyk kapcsolata

5.3. A prototípus funkcionalitása

A prototípus funkcionalitása messze elmarad a koncepcióban létező végleges rendszerétől, de jól kell, hogy reprezentálja a rendszerben rejlő lehetőségeket. Erre a célra a következők elkészítését terveztük:

- Vállalatok, Felhasználók tárolása

- Authentikáció alap implementációja
- Webszolgáltatás kialakítása, vélhetően végleges interfész-definícióval
- Webes adminisztrációs felület prototípusának kialakítása szemléltetés céljára
- Egyszerűbb munkafolyamatok futtatása. (Ciklus nélküli, egyszerű feltételekkel előrehaladó folyamatok)
- Valamely, más projekt keretében elkészült mobil kliens felhasználása

A rendszer középpontjában a Felhasználók (User) állnak. Authentikációnál egyértelműen azonosíthatóak a vállalatukkal, és a felhasználói nevükkel. Egy vállalaton belül nem lehet több azonos felhasználói névvel rendelkező felhasználó, valamilyen módon meg kell különböztetni az azonos valódi névvel rendelkező embereket is.

Minden felhasználónak adhatóak titulusok, illetve csoportok a cégen belül. Ezek segítségével tervezzük szabályozni a felhasználható számára elérhető jogosultságokat (ServerRole). A jogosultságok explicit módon adhatóak, illetve megtagadhatóak a csoport, illetve titulus tagjainak, valamint akár felhasználói szinten is, bár ez utóbbi lehetőség erősen ellenjavallott.

Az autorizáció részletes megtervezését, és implementálását egy későbbi iterációra halasztottuk, mivel nem egyszerű kérdés, és nem szükséges a koncepció működésének demonstrálásához. Az autorizáció elvégzéséhez szükséges mezők, és metódusok elkészültek, de a rendszer folyamatai egyelőre nem veszik ezeket figyelembe, valamint a jogosultságokra is csak minták vannak a rendszerben.

Minden vállalat definiálhat saját munkafolyamat leírókat (WorkflowDescriptor), amelyek alapján felhasználók hozzárendelésével munkafolyamatok indíthatók. A koncepció szerint az indításkor felhasználói csoportok (WorkflowGroup) kerülnek kijelölésre, amely tagjaira a rendszernek megkötései is lehetnek, és a munkafolyamat előrehaladása során további csoportok jöhetnek létre, a munkafolyamat eseményei alapján.

A munkafolyamatok minden lépése egy munkafolyamat-elem leíró (WorkflowItemDescriptor) alapján történik. A jelenlegi prototípus minden elemre tartalmazza, hogy melyik munkafolyamat-csoport tagjainak kell üzenetet küldenie, mi a küldött üzenetnek mi a tartalma, illetve mik a kezdeti, és végfeltételei.

Minden lépésben kiküldésre kerül a felhasználóknak egy üzenet (Message), amely egy űrlap leírását tartalmazza. A kliensek ezt a saját módjukon meg tudják jeleníteni, és a kitöltés végeztével azonos formátumban visszaküldeni. Jelen prototípus tervei szerint a munkafolyamatok léptetését csak a válaszüzenetek (Response) beérkezése válthatja ki. A későbbi tervek között szerepel például időzíthető események hozzáadása is.

A webszolgáltatást a SessionManager, illetve a WorkflowManager osztály támogatja,

amelyek közül az előbbi az autentikációért, autorizációért, illetve a bejelentkezett felhasználók menedzseléséért felel, míg az utóbbi az üzenetek küldését, válaszüzenetek feldolgozását, és a munkafolyamatok kezelését végzi. A CompanyManager osztály feladata a vállalat kéréseinek kiszolgálása, például a felhasználók, vagy a munkafolyamat leírók kezelése.

5.4. A prototípus belső működése

Minden munkafolyamat létrehozásához szükség van egy munkafolyamat leíróra. A leíró tartalmazza a teljes szerkezetét a munkafolyamatnak, minden lehetséges elágazással, feltételekkel, dolgozók csoportjainak a leírásával. Egy workflow létrehozásához szükség van a leírójára valamint a dolgozók csoportjainak prototípusaira – ez tartalmazza a hozzárendelendő felhasználókat. A későbbiekben a csoportok dinamikusan is kialakulhatnak majd, de a kezdeti csoportokra akkor is szükség lesz.

A munkafolyamat indításakor a feltétel nélküli elemek készülnek el a leírójukból, a továbbiak a válaszok beérkezése, vagy a jövőben egyéb események, például egy időzítő lejártának bekövetkezése után indulhatnak el, ha teljesült az indítás feltétele.

Minden munkafolyamat elem egyféle üzenetet küld el a hozzárendelt felhasználói csoportnak, és a beérkező válaszüzenetek, vagy egyéb események alapján fejeződhet be. Egyéb esemény lehet akár egy másik elem befejeződése is, nem feltétlen kell az elemen belül maradni. Annyi a korlátozás a későbbi lehetőségekben, hogy egy munkafolyamat nem hathat egy másikra.

A feltételek egy fa szerkezetben ábrázolhatók. Minden feltételt egy a ConditionBase absztrakt osztályt öröklő objektum reprezentál, amely vagy konkrét feltételt fogalmaz meg, vagy egy elágazást. Az elágazás kiértékelő függvénye a két ágában lévő feltételek igazságtartalmából, és az elágazás típusából állítja össze a visszatérési értékét. Ezáltal lényegében tetszőleges bonyolultságú feltételrendszer létrehozását tesszük lehetővé, bár az adminisztrációs felület átláthatóságának megoldására nem sikerült megoldást találni bonyolult feltételek esetén. A legtöbb esetben nincs is szükség néhány ágnál többre egy feltétel megfogalmazásakor. Egy munkafolyamat lezárása is feltételhez kötött esemény.

Az üzenetekben, illetve válaszüzenetekben az AttachmentBase osztály leszármazottjaiból álló csatolmánylista található. Ezek különféle vezérlőket reprezentálnak, amelyet a kliensalkalmazás a saját eszköztárával kell, hogy megjelenítsen.

A felhasználók autentikációjáért, autorizációjáért, és a bejelentkezett felhasználók nyilvántartásáért a SessionManager osztály felel. A webszolgáltatásnak minden kérés végrehajtása előtt egyeztetnie kell a hozzá tartozó SessionManagerrel, hogy végrehajtható-e az adott utasítás.

A SessionManager jóváhagyása után a munkafolyamatokkal kapcsolatos feladatokat a WorkflowManager osztály hajtja végre. Az üzenetek feldolgozása például az adatbázisba menti a megkapott üzenetet, és ellenőrzi, hogy annak hatására történik-e valami a munkafolyamatban. Ha igen, akkor végre is hajtja a léptetést.

A szerver minden kommunikációt JSON sorosított objektumokkal valósít meg. Még a munkafolyamat leírók leírása is JSON dokumentumként van elküldve, illetve az üzenetekhez hasonlóan az adatbázisban is sorosítva vannak tárolva. Nem tartjuk kizártnak a BPEL – BPMN^[15] támogatás implementálását a jövőben, de a rendszer céljai szempontjából nehézkes, bonyolult lenne a felhasználása, egyszerűbbnek tartottuk a saját objektumaink sorosításával megoldani a munkafolyamatok leírásának a kérdését.

6. Az MFlow munkafolyamat szerkesztő

Az MFlow projekthez készült munkafolyamat szerkesztő arra szolgál, hogy az ügyfelek a szerveren tárolt munkafolyamat-leírókat interaktív módon szerkeszthessék. A 7. ábrán látható a munkafolyamat szerkesztő felület prototípusának megjelenése.



7. ábra A munkafolyamat szerkesztő

6.1. Elvek és motiváció

Alábbiakban olvasható a munkafolyamat szerkesztő mögötti motiváció, és az, hogy miért is van szükség erre a komponensre.

6.1.1. Háttér

A munkafolyamatok leírása és definiálása általában olyan feladat, amit fejlesztők végeznek. Legtöbb ezen feladatra tervezett eszköz vagy keretrendszer nem rendelkezik olyan felhasználói felülettel, amely könnyen átlátható, és amelyet végfelhasználók is szerkeszteni tudnak.

6.1.2. Célunk

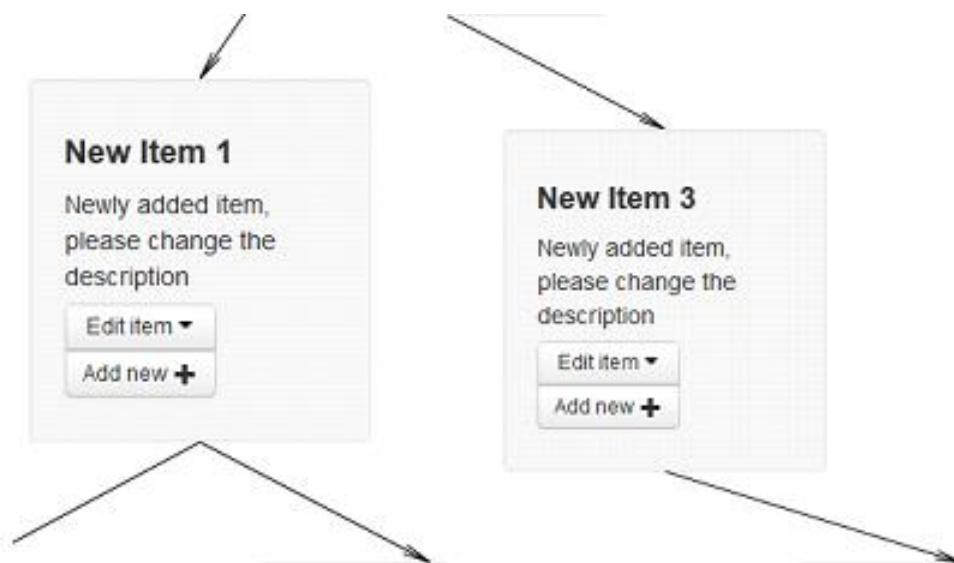
Az MFlow projekt egyik célja, hogy a kis- és középvállalkozások igényeihez igazodjon. Ezen vállalkozások általában nem engedhetik meg maguknak, hogy szoftverfejlesztőket foglalkoztassanak arra célra, hogy a munkafolyamatokat összerakják, mert az esetek többségében nem tudnák megfizetni.

Célunk ezért az, hogy egy olyan felhasználói felületet biztosítsunk, amelyet egy végfelhasználó is könnyen átlát és mely segítségével a szoftverfejlesztésben járatlan személyek is képessé válhatnak arra, hogy vállalatok munkafolyamatait modellezzék.

6.1.3. Építőelemek

Munkafolyamatainkat az MFlow projektben munkafolyamat-leírókkal modellezhetjük. Ezen leírók alapján történik a konkrét munkafolyamatok működése. Az egyes munkafolyamatok és leíróik viszonya hasonló az objektumorientált programozásban levő osztály és objektum fogalmak viszonyához.

A munkafolyamat -leírók munkafolyamat-elemeleírókból állnak. Ilyen leírók láthatóak a 8. ábrán.



8. ábra Munkafolyamat elemek a szerkesztőben

Ezen leírók egy-egy kisebb egységet, vagyis a teljes munkafolyamat egyes részeit jelképezik. Ezek felelősek egy bizonyos konkrét feladat végrehajtásáért. A munkafolyamat-elemek közötti kapcsolatokat nyilakkal jelöljük. Ha egy munkafolyamat elindulása egy másiktól valamilyen feltétel szerint függ, egy nyilat húzunk közéjük, amely a munkafolyamat indulásának irányát mutatja meg.

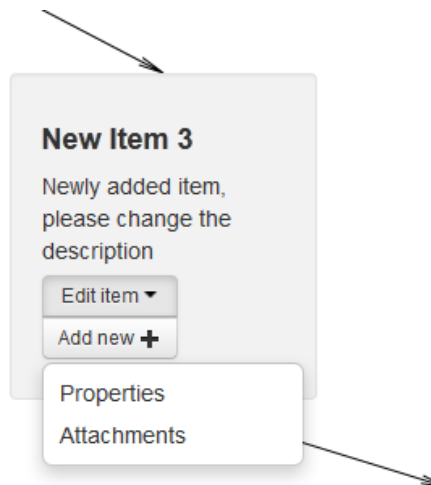
6.2. Felhasználói felület leírása

A felhasználói felület célja, hogy könnyen kezelhető és jól átlátható legyen. Fontos megemlíteni, hogy a prototípusban megtalálható szerkesztő még nem teljes, azonban a cél a koncepció és az elvek bemutatása.

Fentieknek megfelelően a felületen egyszerű "fogd és vidd" (drag and drop) módszerrel tetszőlegesen helyezhetjük el a munkafolyamat-elemleírókat. Nyíl mutat két leíró között akkor, ha az egyik indulásának feltétele a másik befejeződése.

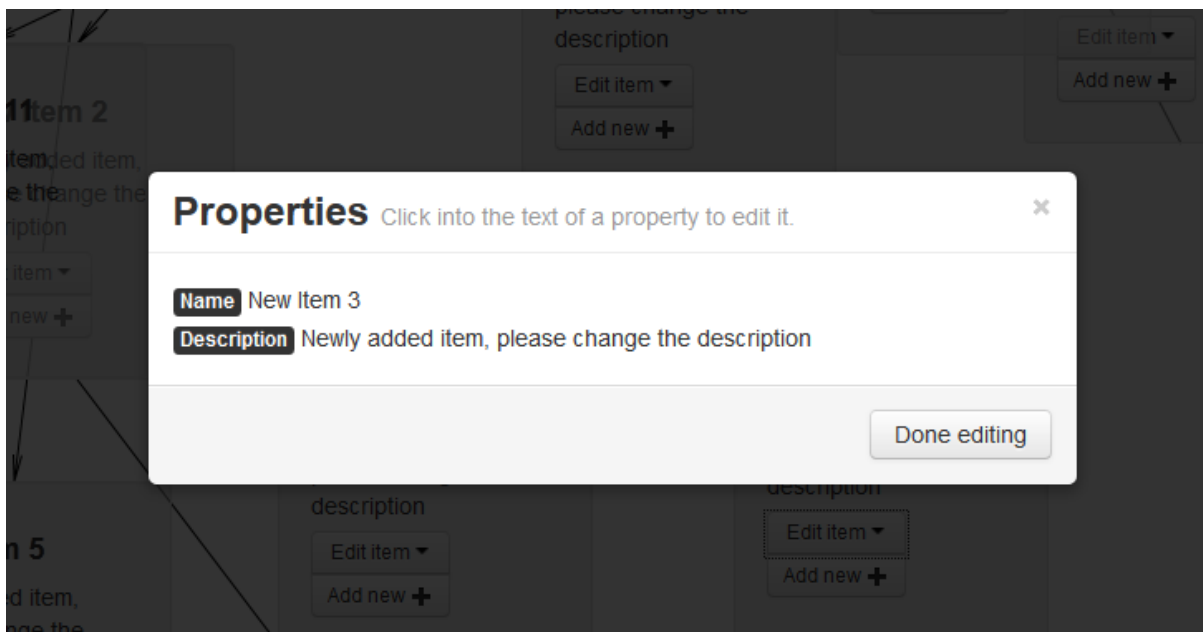
Új elemet az egyes elemeken, vagy a felület tetején található gombokkal adhatunk hozzá. Előbbi esetben az új elem indulásának automatikusan feltétele lesz annak befejeződése, amelyiken a gombot megnyomtuk.

Egyes elemeket a rajtuk található szerkesztés gombra kattintva szerkeszthetünk. Ez a gomb egy menüt jelenít meg, amelyen több szerkesztési lehetőség közül választhatunk. A 9. ábrán látható ez a gomb. A „properties” menüben az elem nevét, leírását szerkeszthetjük, míg az „Attachments” menü az elemhez tartozó űrlapot lehet szerkeszteni.



9. ábra Munkafolyamat elem szerkesztése

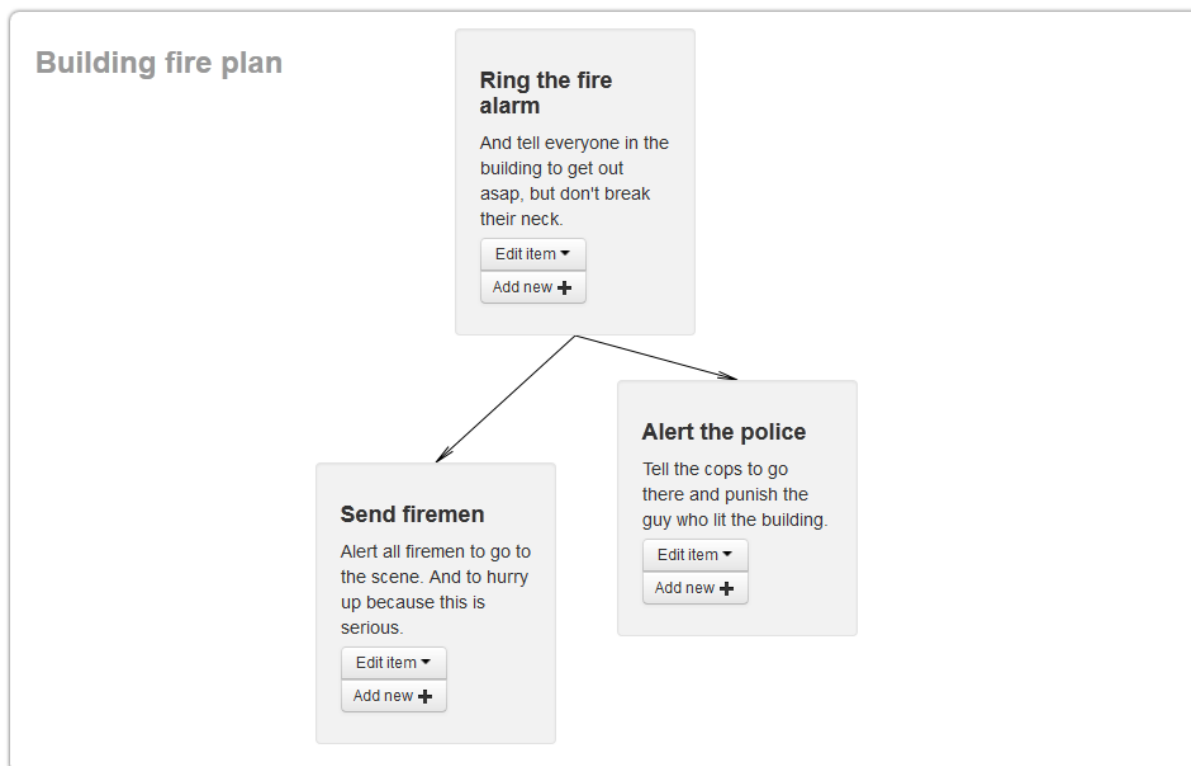
A menü egyes elemeire kattintva az oldalon belül külön ablak (de nem új böngészőablak) nyílik meg, amelyen az adott elemhez tartozó beállításokat szerkeszthetjük. Ez a 10. ábrán látható.



10. ábra A munkafolyamat elem tulajdonságai

Az egyes tulajdonságok szerkesztése szimplán a szövegbe való kattintással működik. Ha a tulajdonság szövegébe kattintunk, akkor megjelenik egy kurzor, és a szövegszerkesztő programokban megszokott módon szerkeszthetjük az értékeket.

A 11. ábrán látható a szerkesztőfelület a "demó munkafolyamat" megjelenítése közben.



11. ábra Demó munkafolyamat

A demó munkafolyamat a szerkesztő fejlesztése során volt segítségünkre, illetve jól modellez egy példa munkafolyamatot.

Példa munkafolyamatunk egyszerű, háromelemű, és egy épület tűzkatasztrófa esetén történő kiürítésének tervét hivatott modellezni. Három eleme a következő:

- Amennyiben tűz van, megszólaltatni a riasztót
- Felhívni a tűzoltóságot
- Szólni a rendőrségnek, hogy tartóztassák le azon személyt, aki felgyújtotta az épületet

Ez a példa egyben szemlélteti a munkafolyamat csoportok szerepét is, a három elem ugyanis három különböző csoportnak szól:

- Épületben dolgozók, akik jelezhetik, ha tűz van
- Tűzoltók, akik kimennek a helyszínre
- Rendőrök, akik részt vesznek a nyomozásban

6.3. Felhasznált technológia

Az MFlow projekt technológiai választéka egyedülálló; több webes technológiai újdonságot kombinál forradalmian új módon. Alább olvashatóak a technológiai döntés részletei.

6.3.1. Háttér

Amikor kiválasztottuk, hogy milyen technológiát használjon az MFlow, a szempont az volt, hogy a választott technológia segítségével modern felhasználói felületet hozhassunk létre, amely azonban mégis a lehető legtöbb lehetséges ügyfél számítógépén fut.

Jelenleg a legkönnyebben nagy számú felhasználói réteg elérésére a web használható. A különféle platformokon futó web böngészők viszonylag konzisztens API-t nyújtanak, ezért a választás arra esett, hogy a szerkesztő alkalmazás böngészőkben fusson.

Tovább vizsgálva a lehetséges technológiákat, arra jutottunk, hogy a böngészőkbe épülő modulok (vagy plugin-ek) működése nem biztosított minden platformon, továbbá túl inkonzisztens és az ilyen modulok telepítésének megkövetelése nehezítené a felhasználói bázis növekedését.

A fentiek alapján arra jutottunk tehát, hogy a lehető legtöbb böngészőben konzisztensen működő HTML 5 és JavaScript alapú megoldás fejlesztését választjuk.

6.3.2. Racionalitás

2012-ben a HTML 5 és a JavaScript általunk használt részhalmaza már minden böngésző programban elérhető, legalább a legújabb verzióban, de jelentősebbek esetén több évre visszamenőleg is jelen voltak már a kérdéses funkciók. Tapasztalatunk szerint a kis- és középvállalkozások számára nem jelent gondot egy elég korszerű böngésző telepítése és használata, ezért nyugodtan kihasználhatjuk az ezen technológiák által nyújtott előnyöket.

6.3.3. Előnyök

A választott technológia előnye, hogy sok konkurencsallal ellentétben nem igényli az oldal folyamatos újratöltését. Mindössze egy letöltési művelet szükséges az oldal betöltődésekor, amelyet a felhasználó által kiváltott tetszőleges számú mentési művelet követhet. Ez sávszélességet is spórol, valamint a felhasználók számára is kevesebb megszakítással jár

További előny, hogy, mivel a feldolgozás nagy részét így a kliensoldalra hárítja a szoftver, ezért a szerver terhelése, számításigénye, és ezáltal fenntartási költsége alacsonyabb lehet.

Bár a fentiek már önmagukban is jelentősek, mégis fontos kiemelni, hogy napjaink webes technológiája jelentősen magas fokú felhasználói élmény nyújtására is alkalmas.

6.3.4. Felhasznált megoldások (kliens oldal)

jQuery

Kliensoldali JavaScript keretrendszer, mely a DOM (dokumentum-objektummodell) manipulálására és eseménykezelésre kínál kényelmesen használható és jól kiegészíthető API-t.

Mivel ez az alapja több egyéb keretrendszernek, amelyet használunk, ezért természetes, hogy az MFlow is ezt használja a konkurensok helyett^[17].

jsRender

jQuery alapú függvénykönyvtár, amely kliens oldali sablonok, vagy más szavakkal kliens oldalon történő HTML generálásra ad nagyon jól használható megoldást.

Ennek köszönhetően levettük a szerver válláról a HTML generálás terhének egy részét. Lehetővé válik az, hogy az aszinkron hívások esetében ne a szervernek kelljen a választ HTML formátumban visszaadnia, hanem a kliens egy JSON (JavaScript Object Notation) objektumot kap vissza, amelyből a megjelenítést szükséges HTML kódot helyben állítja elő^[18].

Twitter Bootstrap

Felhasználói felületek építésére szolgáló keretrendszer, melyet a Twitter fejleszt. Egyaránt tartalmaz JavaScript és CSS kódot. Jó minőségű alapértelmezett megjelenést biztosít a komponensek számára, és könnyen testre szabható.

jQuery UI

Szintén felhasználói felületekhez tartalmaz komponenseket, de az MFlow projekt csak a "fogd és vidd" implementációt használja belőle.

HTML 5

A felhasználói élmény maximalizálása érdekében az MFlow workflow szerkesztő a HTML 5 specifikáció egyes elemeit használja. A canvas (vászon) technikát az elemek összekötésére szolgáló nyilak rajzolására használjuk, a contenteditable-t pedig a kényelmes szerkesztéshez.

6.3.5. Felhasznált megoldás (szerver oldal)

Szerver oldalunkon a Microsoft ASP.NET technológiáját használjuk, ennek egyes elemeit külön funkciókra. Alább ennek részletezése olvasható.

ASP.NET MVC

A Microsoft-féle MVC (Model-View-Controller) keretrendszer az ASP.NET-hez. A fejlesztés koncepcionálisan jobban illik a webes fejlesztéshez. Megjegyzés: az MVC architektúra részletezése túlmutat jelen dokumentum keretein.

ASP.NET WEB API

Aszinkron HTTP kérések hatékony kezelésére kiválóan alkalmas. Gyakorlatilag közvetlen hozzáférést és automatikus szerializációt biztosít. Ennek bővebb leírását lásd a működésről szóló fejezetben.

6.4. Működés

Az MFlow workflow szerkesztő működése a hagyományos AJAX modellhez hasonlít, de bizonyos részekben és elvekben eltér tőle.

A hagyományos AJAX megoldás a következőképpen megy.

1. A kliens küld egy aszinkron HTTP POST kérést a szerver felé
Ebben az esetben az aszinkron alatt azt értjük, hogy a felhasználó számára érzékelhetetlen módon történik, vagyis nem okoznak fennakadást a felhasználói élményben. A felhasználó tovább használhatja az oldalt közben.
2. A szerver feldolgozza kérést, és küld választ XML formátumban
3. A kliens feldolgozza a válasz XML-t, és esetlegesen módosításokat hajt végre a felhasználói felületen.

Mivel a kliensoldalon ezt általában JavaScript nyelven implementálják, ezért a megoldás az AJAX (Asynchronous JavaScript and XML) nevet kapta.

Az MFlow ugyanezt a mechanizmust használja, azonban néhány jelentős különbség van.

- Az MFlow az aszinkron lekérések esetén a HTTP protokolt szemantikusan, a REST elveknek megfelelően használja. Vagyis elemek letöltéséhez GET, újak létrehozásához POST, és létezőek frissítéséhez PUT metódust használ.
- Az MFlow a hagyományos XML alapú adatkommunikáció helyett JSON formátumot használ. Ennek nagy előnye, hogy problémamentesen konvertálható natív JavaScript objektummá is. Így gyakorlatilag a nyelv szempontjából objektumokat küld a szerver, semmiféle komoly deserializálás vagy értelmezés nem kell.

A workflow szerkesztő működési folyamata

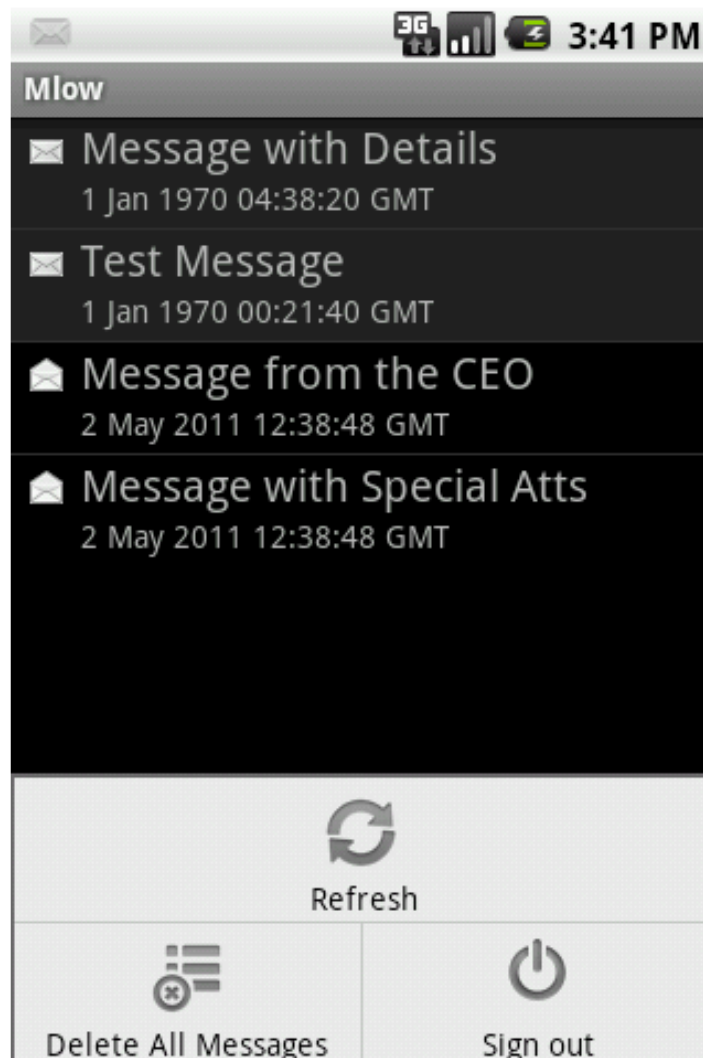
Az MFlow HTML oldalai a Razor view engine segítségével vannak elkészítve. Miután a HTML betöltődött, a kliens elkezd futtatni a workflow szerkesztő JavaScript kódját.

Ennek folyamányaként a következő folyamat zajlik le:

- A kliens aszinkron lekérést indít (HTTP GET) és letölti a szükséges leíró, amelyet JavaScript objektumként értelmezni is tud és lementi a memóriába
- Kliensoldali HTML sablonok segítségével megfelelő megjelenést állít elő a letöltött objektum számára, és azt megjeleníti a felhasználó felé.
- A felhasználó tetszése szerint módosításokat hajt végre, vagyis szerkeszti a leíró. A módosítások a memóriában történnek, és folyamatosan megjelennek a felhasználói felületen is. Vagyis az alkalmazás a kliens memóriájában tartja nyilván a módosításokat.
- Amikor a felhasználó megnyomja a mentés gombot, akkor a kliens a memóriájában tárolt objektumot egy HTTP PUT vagy POST kérés segítségével felküldi a szerverre, amely értelmezi és tárolja azt.

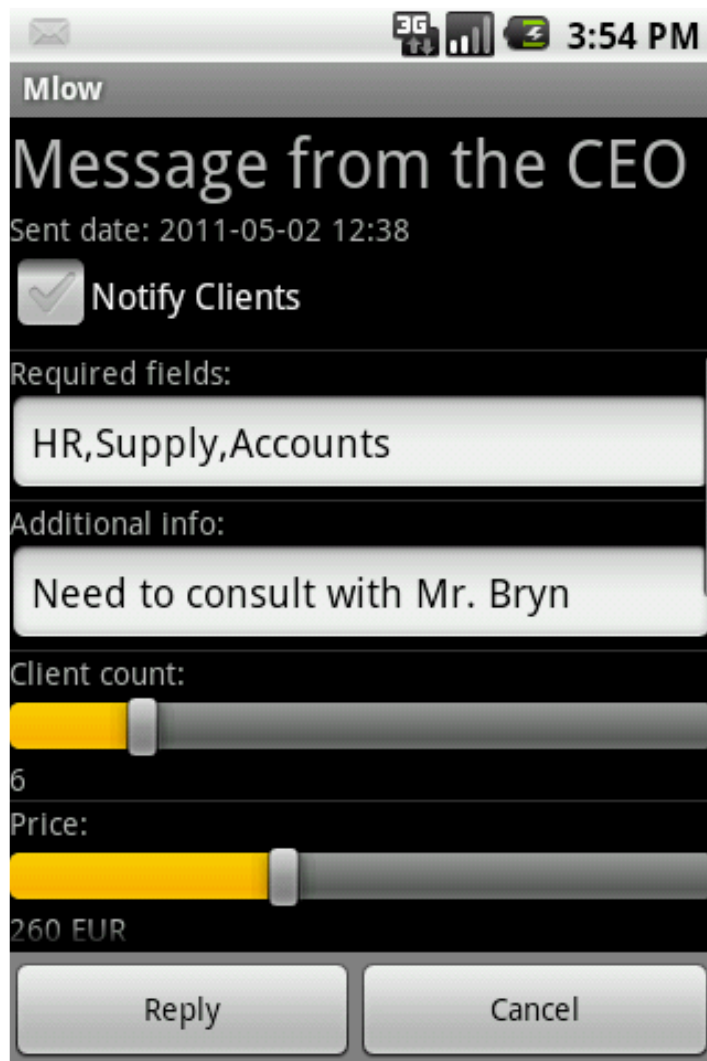
7. Mobil kliens minta

Egy másik projekt keretében elkészült egy kliens Androidos okostelefonokra. Sajnos ez az alkalmazás a webes interfész jelenlegi verziójával nem kompatibilis, de a felülete a frissítés után is változatlan lesz.



12. ábra Az androidos kliens aktuális üzeneteket listázó felülete

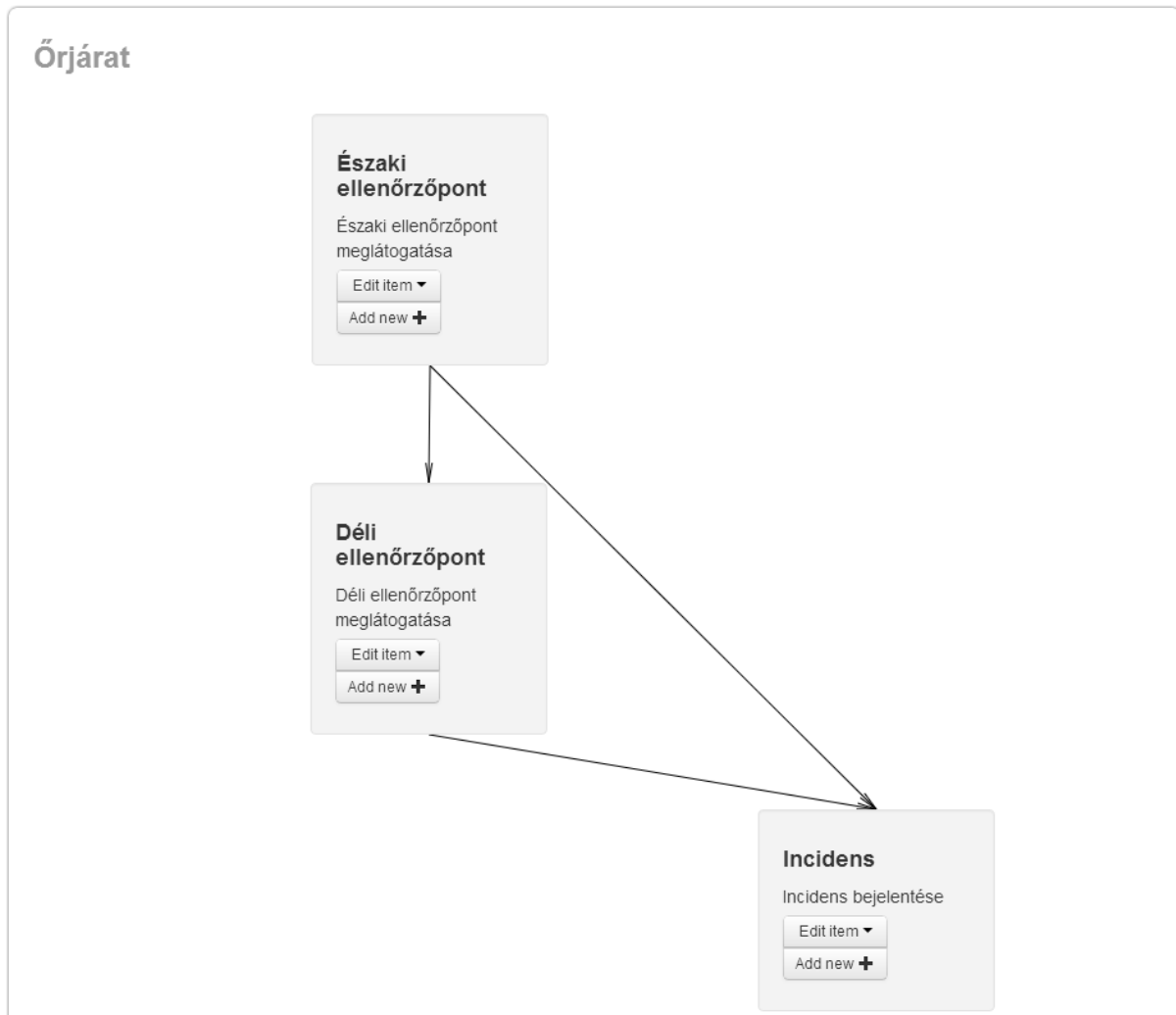
A 12. ábrán látható az Androidos kliens üzeneteket listázó felülete. Innen valamely üzenetet kiválasztva az üzenet-nézetre jutunk, amely egy listában adatkötés segítségével jeleníti meg az üzenetben található csatolmányokat. Ezt a 13. ábrán láthatjuk. Vegyük észre, hogy a vezérlők az Android operációs rendszeren megszokott módon jelennek meg. Ugyanígy, például az iOS, vagy Windows Phone kliensek is a natív vezérlőket veszik majd igénybe az űrlapok megjelenítéséhez, ezáltal is kényelmesebbé téve az alkalmazás használatát. A webszolgáltatás lehetővé teszi a munkafolyamatok lekérését is, hogy az üzenetek ez alapján csoportosíthatóak legyenek, de ez új fejlesztés, a klienst készítőknél még nem állt rendelkezésére ez a szolgáltatás^[16].



13. ábra Az androidos kliensen megjelenő űrlap

8. Használati példák a rendszerben

8.1. Biztonsági őr



14. ábra Őrjárat munkafolyamat

Az ismertett biztonsági őrök alkalmazó vállalat a 14. ábrán láthatóhoz hasonló munkafolyamatok segítségével tudná megvalósítani a feladatot. Természetesen ellenőrzőpontból bármennyi lehet, és bárhogy lehet őket nevezni. Az incidens ágat is tetszőlegesen lehet bővíteni, például több őr kiküldésével, a rendőrség kihívásával, és ennek eredményével.

Az ellenőrzőpontok elemeiben egy checkbox található, amely incidenst jelez, valamint egy automatikusan kitöltött GPS-koordináta mező. Az incidens elem szöveges leírást, fényképet vár, és lehet rajta keresztül erősítést kérni, vagy jelezni, hogy ki lett hívva a rendőrség.

Egy jövőbeli verzióban egy ilyen munkafolyamat beágyazható lenne egy másik, ezeket időzítetten, automatikusan indító folyamatba.

8.2. Futárszolgálat



15. ábra Futárszolgálat munkafolyamat

A futárszolgálat-példa egyszerűsített munkafolyamata látható 15. ábrán. A csomag leadása egy későbbi verzióban több próbálkozást is támogathat, ciklus felhasználásával. Az automatikus munkakiosztás megvalósulása után a feladatot elvállalók közül a rendszer nem az elsőnek, hanem például a kiindulási ponthoz legközelebb eső futárnak adhatja oda a feladatot.

Hasonló munkafolyamat tervezhető az utazó ügynök, és a külső helyszínek munkafolyamatai számára is.

9. Összefoglaló

9.1. A jelenlegi állapot

Ezen dolgozat létrejötté egy hosszabb folyamat eredménye, és több projekt is kapcsolódik hozzá – például a kliens alkalmazások elkészítését célul kitűzők. A cél már a projekt elején adott volt, de az eléréséhez vezető úton számos kérdés került megvitatásra. Többször is kiderült, hogy egy korábban kiválasztott technológia mégsem megfelelő a számunkra. Szerencsére komoly változtatásra csak egy alkalommal került sor, épp a webes interfész

implementációjának kapcsán. Ennek a megoldása során vált inkompatibilissé az androidos kliens a szolgáltatással.

Az MFlow szoftver koncepciója a munkafolyamat kezelő alkalmazások körében egyedi megközelítést ad, és már a mostani prototípuson is úgy látjuk, hogy van benne potenciál. Világos célokat fogalmaztunk meg, és már a prototípus implementációja is minőségi szoftverfejlesztés eredménye. Mivel már a korábbi prototípus bevezetésén is látszik, hogy szükség, és igény van a mobil munkafolyamat kezelésre, ezért mindenképpen jónak tartjuk a fejlesztési irányt, és érdemesnek tartjuk a jövőben is foglalkozni vele. A jövőbeli célok is elég élesen körvonalazódtak, és mivel a minőségi szoftverfejlesztést központi kérdésnek tartjuk, a karbantarthatóság, és későbbi ötletek beillesztése sem okozhat problémát.

9.2. Továbbfejlesztési lehetőségek

A jövőbeli terveink között szerepel a szofisztikált munkafolyamatok támogatásának implementációja, ciklusokkal, időzített, és feltételesen végrehajtott, automatizált eseményekkel támogatva. Implementáljuk az autorizációt, konzultálva potencionális ügyfelekkel, az általuk megfogalmazott igények szerint, és az adminisztratív funkciók kibővítését is.

Megoldást szeretnénk nyújtani a munkafolyamatok kivételkezelésére, hogy a megfelelő jogosultságokkal kivételes esetben bele lehessen avatkozni egy munkafolyamat lefutásába. Egy ilyen szolgáltatás nagyban növelné a rendszer alkalmazhatóságát a valóságban midnenféle trükközés nélkül is (a jelenlegi implementációban is lehetséges a kivételkezelés, ha minden munkafolyamathoz hozzáadunk egy kivételkezelő ágat, de ez nem hatékony, és nem felhasználóbarát.)

A vállalat vezetői számára lehetőséget szeretnénk kínálni, hogy az adminisztratív felületen összefoglalót láthassanak a munkafolyamatokról, illetve kereshessenek közöttük, ezáltal lehetőséget biztosítva a céges folyamatok hatékonyságának vizsgálatára.

Előttünk áll a rendszer felügyeleti moduljának elkészítése, amely segítségével a rendszerben történő események naplózhatóvá válnának, valamint a rendszer állapotáról információ lenne lekérhető az üzemeltetők által.

Az elterjedőben lévő, vagy jövőbeli mobil technológiák támogatása is fontos szempont, például a telefonokban kezd terjedni az RFID technológia, amelynek a használata például a biztonsági őr példa integrációját megvalósíthatja a beépíthető biztonsági rendszerekhez.

10. Forrásgyűjtemény

[1] <http://blogs.iod.com/2010/05/19/the-biggest-problems-facing-small-businesses-today/>
<http://www.dynamicbusiness.com.au/finance-cash-flow/tips-for-identifying-pain-points-in-workflows-07052012.html>

Személyes tapasztalat

[2] <http://www.microsoft.com/hun/dynamics/crm/>

- [3] <http://www.microsoft.com/biztalk/en/us/default.aspx>
- [4] <http://www.processmaker.com/>
- [5] http://www.iso.org/iso/home/standards/management-standards/iso_9000.htm
<http://www.mszt.hu/tanusitas/mir.html>
- [6] <http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA263403>
A Szoftvertechnológiák című tárgy
- [7] <http://www.processmaker.com/customer-solutions>
<http://www.microsoft.com/biztalk/en/us/case-studies.aspx>
- [8] <http://www.microsoft.com/biztalk/en/us/pricing-licensing.aspx>
- [9] Novák Péter – Android kliens fejlesztése elosztott workflow rendszerhez
(Szakdolgozat)
- [10] <http://nhforge.org/>
<http://www.fluentnhibernate.org/>
- [11] <https://www.windowsazure.com/>
- [12] <http://visualstudiogallery.msdn.microsoft.com/b0e0b5e9-e138-410b-ad10-00cb3caf4981>
<http://msdn.microsoft.com/en-us/library/gg597391.aspx>
- [13] <http://www.asp.net/mvc/mvc4>
- [14] <http://james.newtonking.com/projects/json-net.aspx>
- [15] <http://www.bpmn.org/>
- [16] Novák Péter – Android kliens fejlesztése elosztott workflow rendszerhez
(Szakdolgozat)
- [17] <http://jquery.com/>
<http://docs.jquery.com/>
- [18] <http://msdn.microsoft.com/en-us/magazine/hh882454.aspx>
<http://msdn.microsoft.com/en-us/magazine/hh975379.aspx>
- [19] <http://twitter.github.com/bootstrap/index.html>
- [20] <http://jqueryui.com/>
<http://api.jqueryui.com/>
- [21] <http://www.w3.org/TR/2011/WD-html5-20110525/>