

Bátfai Samu

Egy testetlen fejlődési robotikus ágens

Ed. SAMU SRS, v. Nahshon.0.0.3

Copyright © 2015 Dr. Bátfai Norbert

Samu Bátfai (Nahshon) Software Requirements Specification

Copyright (C) 2014, Norbert Bátfai. Ph.D., batfai.norbert@inf.unideb.hu

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Ez a program szabad szoftver; terjeszthető illetve módosítható a Free Software Foundation által kiadott GNU General Public License dokumentumában leírtak; akár a licenc 3-as, akár (tetszőleges) későbbi változata szerint.

Ez a program abban a reményben kerül közreadásra, hogy hasznos lesz, de minden egyéb GARANCIA NÉLKÜL, az ELADHATÓSÁGRA vagy VALAMELY CÉLRA VALÓ ALKALMAZHATÓSÁGRA való származtatott garanciát is beleértve. További részleteket a GNU General Public License tartalmaz.

A felhasználónak a programmal együtt meg kell kapnia a GNU General Public License egy példányát; ha mégsem kapta meg, akkor tekintse meg a <http://www.gnu.org/licenses/> oldalon.

<http://gnu.hu/gplv3.html>

COLLABORATORS

	<i>TITLE :</i> Bátfai Samu		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Bátfai, Norbert	2015. december 13.	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME
Nahshon.0.0.1	2015-11-28	Initial document release for comment.	
Nahshon.0.0.2	2015-12-02	Rendszermodellek és rendszer-evolúciós részek vázlatos kidolgozása.	
Nahshon.0.0.3	2015-12-04	A SAMU-OOCWC felvetése és javítások.	

Tartalomjegyzék

1. Szójegyzék	1
2. Bevezetés	3
2.1. Bátfai Samu	3
2.2. A fejlesztendő rendszer áttekintő leírása	4
3. Samu szoftverkövetelményei	5
3.1. Samu áttekintése	5
3.2. Követelmények	5
3.2.1. Funkcionális követelmények	5
3.2.2. Nem funkcionális követelmények	5
3.2.3. Szakterületi követelmények	6
3.2.3.1. Q-tanulás	6
3.2.3.2. A kutatási feladatok aspektus-orientált megfogalmazása	6
3.2.4. Felhasználói követelmények	6
3.2.5. Rendszerkövetelmények	7
3.2.5.1. Használati esetek	7
3.2.6. Interfészek	8
3.3. Összefoglalás	8
4. Rendszermodellek	9
4.1. Rendszer-specifikáció	9
4.1.1. Samu viselkedése madártávlatból	9
4.1.2. Samu adatmodelljei	10
4.1.3. Samu objektummodellje	10
5. Samu GUI	11
5.1. Igény egy grafikus felhasználói felület készítésére	11
5.2. Kezelés, a felület kinézete	12
5.3. A fejlesztés lépései nagy vonalakban	12
5.4. Készítette	13

6. Rendszer-evolúció	14
6.1. Samu vezérlése	14
6.2. Beágyazott Samu	14
7. Irodalomjegyzék	16
7.1. Kutatási cikkek	16
8. Tárgymutató	17

Ábrák jegyzéke

3.1. Samu Q-tanulása	6
3.2. Samu használati esetei	7
3.3. A beszélgetés és a tanulás kapcsolata	8
3.4. A tanulás	8
4.1. Samu állapotdiagramja	9
4.2. Samu osztálydiagramja	10
5.1. Samu GUI tanulás közben	12

Táblázatok jegyzéke

4.1. Az állapotok leírása	9
4.2. Az állapotátmenetek leírása	10

1. fejezet

Szójegyzék

C

CUDA [CUDA]

Olyan az NVIDIA GPU-kon futó extrém párhuzamos programozási platform, amely adott esetekben PC-n megvalósított HPC számítási teljesítményt tesz elérhetővé.

I

IEEE Recommended Practice for Software Requirements Specifications (IEEE, 1998) [IEEE Std 830-1998]

A szoftverkövetelmény dokumentum [IEEE szabványa](#). A jelen dokumentum a Samu projekt SRS dokumentuma.

D

Developmental Robotics [DevRob]

A robotika azon interdiszciplináris (például robotika, informatika, neveléstudomány, pedagógia, pszichológia, orvostudomány, nyelvészet stb.) ága amely az MI fejlesztésben irányadónak tartja az emberi egyéni fejlődésének tanulmányozását.

Deep Q-Learning [DeepQ]

A megerősítéses tanulás Q-learning technikájának mély neurális hálózati implementációkkal támogatott megvalósítása.

DocBook XML [DB]

A [DocBook](#) elsősorban az informatikai tárgyú írásos dokumentumok készítésére és annotálására kifejlesztett modern, szabványos jelölőnyelv.

L

Liv-Zempel-Welch algoritmus [LZW]

Széles körben használt, gyors és hatékony (például gif képek vagy gzip programok) tömöríti algoritmus.

M

Multilayer Perceptron [MLP]

Rejtett rétegeket tartalmazó mesterséges neurális hálózat, ahol minden neuron egy perceptron.

N

Natural Language Processing [NLP]

Az informatika azon ága, amely a természetes nyelvek feldolgozásával foglalkozik, különös tekintettel annak gépi feldolgozására, az ember-gép kapcsolat ilyen alapú interakciójára.

U

The Yearbook of the Programmers of University of Debrecen [UDPROG]

A Debreceni Egyetem [reguláris programozás oktatásához](#) kapcsolódó, Facebook-en szervezett [fejlesztői közösség](#), melynek egyik gyűjtőpontjában [szoftverek](#), a másikban maga az [évkönyv](#) áll. A jelen dokumentum, mint egy példa alkalmazás, is bemutatásra kerül ebben a programozói közösségben is.

O

Robocar World Championship (rObOCar World Championship, Robotautó Világversenysorozat)[OOCWC]

A Robotautó Világversenysorozat egy olyan kezdeményezés, amely szoftverplatformot kíván biztosítani az okos városok és a robotautók kommunikációjának kutatásához, alapja az <https://github.com/nbatfai/robocar-emulator>.

2. fejezet

Bevezetés

Ez a dokumentum a Samu projekt SRS (Software Requirements Specification) dokumentuma.

2.1. Bátfai Samu

Samu egy kutatási kezdeményezés egy olyan testetlen fejlődési robotikus (developmental robotic) ágens létrehozására, amely az emberrel összemérhető képességű csevegő robotok létrehozásának alapjául szolgálhat.

Az aktuális kutatási eredményeket a [SAMU] kézirat mutatja be.

A családnév szerepeltetése Samu fejlődési robotikus jellegének ad hangsúlyt, mivel elsősorban családi körben lesz nevelendő!

A jelen pillanatig számos eldobható gyors prototípust fejlesztettünk Samu teszteléséhez, ezek a következők:

- Samu, <https://github.com/nbatfai/samu>
- Isaac, <https://github.com/nbatfai/isaac>
- Jacob, <https://github.com/nbatfai/jacob>
- Judah, <https://github.com/nbatfai/judah>
- Super-Judah, <https://github.com/nbatfai/super-judah>
- Hezron, <https://github.com/nbatfai/hezron>
- Ram, <https://github.com/nbatfai/ram>
- Amminadab, <https://github.com/nbatfai/amminadab>
- Nahshon, <https://github.com/nbatfai/nahshon>
- Salmon, <https://github.com/nbatfai/>

Ezeket az eldobható gyors prototípusokat részletesen a [SAMU] kézirat mutatja be, működésük elvi vázlata a következő: Samu

- olvas/hall egy mondatot (fontos, hogy itt csak sima karakteres input van, tehát az „olvass”, „hall” egyelőre szimpla metaforák, számunkra csak a karakteres be/kimenet a lényeg, praktikusán gondolj a Turing tesztre vagy a Loebner díjra.)
- egy bekezdés mondatait szimulációs programnak kezeli, amely programot futtat, s annak kimenetét egy képre rajzolja
- a képet benyomja egy Q-tanulási algoritmusnak, ahol akkor kap pozitív megerősítést, ha a szimulációs program kimenete és a valóság egyezést mutat, a Q-tanulás kimenete a programírás finomítása

amelyekből az alábbi konkrét lépéseket valósítják meg:

- olvas egy mondatot
- NLP (Natural Language Processing) eszközökkel kibányássza belőle a ALANY-ÁLLÍTMÁNY-TÁRGY (SPO) tripleteket
- amelyeket OO szemléletben egyszerű ALANY.ÁLLÍTMÁNY(TÁRGY); alakú programutasítás formájában ráírja egy képre (lásd a futtatási könyvtárban a sok generált samu-*.png képeket – jelen pillanatban így fest Samu vizuális képzelete)
- a képet benyomja egy Q-tanulási algoritmusnak, ahol akkor kap pozitív megerősítést, ha jól tudja kitalálni a beszélgetés következő mondatát (amit nyilván akár ki is mondhat – azaz írhat ugye – ha csevegésre kerül a sor), a Q-tanulás kimenete egyszerűen a prediktált következő mondat (triplet).

Ezen megvalósítás vizuális interpretációját mutatja a **Samu tanulási algoritmusának** folyamatábrája.

2.2. A fejlesztendő rendszer áttekintő leírása

Samu szoftvere alapvetően egy csevegőprogram, amely lehetőséget ad különböző gépi tanulási algoritmusok kipróbálására, tesztelésére, fejlesztésére. Samu szülei (gondozói) az a szűk családi kör, amelyben a robot fejlesztése történik. A gondozók tipikusan a konzolon keresztül érintkeznek a programmal. Az egyéb felhasználók hálózaton keresztül tudnak a csevegésbe bekapcsolódni. Fontos további szempont hogy adott Samu ágensek tudásukat képesek legyenek egymással hálózaton keresztül megosztani, azaz kvázi egymást tanítani. Illetve Samu tudjon magában is beszélni. Samu programja legyen képes állandóan futni, ha nincs sem kozolos, sem hálózati aktivitás akkor megfelelő korpuszok olvasásával tanítsa magát, illetve önmagával folytasson dialógust! További fontos aspektus, hogy Samu egyben egy kutatási projekt is, ennek megfelelően a kapcsolódó kutatásokat is támogatnia kell a fejlesztendő rendszernek.

3. fejezet

Samu szoftverkövetelményei

3.1. Samu áttekintése

Az előző fejezet rövid szöveges áttekintő leírása alapján elkészítünk egy az UML komponens diagramok jelöléseit használó magyarázó ábrát.

3.2. Követelmények

A szokásos funkcionális és nem funkcionális követelményeken túl Samu esetén meg kell fogalmaznunk fontos szakterületi követelményeket is, hiszen a fejlesztés mögött a mély Q-tanulás alkalmazása és kutatása áll. A szakterületi követelmények után a tervezési folyamatot a rendszerkövetelmények áttekintésével segítjük.

3.2.1. Funkcionális követelmények

Samu alapvető működésével kapcsolatban az alábbi alapvető követelményeket fektetjük le.

- A Samut családi körben elérő gondozók (logikailag) külön csatornákat tudjanak használni, hogy Samu meg tudja különböztetni őket. A családi kör kitüntetett gondozójától Samu fenntartások nélkül tanuljon, azaz korlátlan bizalommal legyen a szülő felé, tehát a szülői bemenetet tekintse igaznak. Legalábbis fiatal korában, ahogyan a humán gyerekekre is ez igaz.
- A Samut felkereső internetes csevegőktől Samu ne, vagy csak nagyon korlátozott mértékben tanuljon.
- A szülő által megadott más kitüntetett Samu programoktól tanulhasson olyan tulajdonságokkal mintha a gondozóktól tanulna.
- Samu programja soha ne álljon le, ha mégis, akkor mentse el állapotát (ez a lementett állapot Samu lelke, maga a futó Samu folyamat a teste) és legközelebb ebből az elmentett állapotból induljon el.

3.2.2. Nem funkcionális követelmények

Samuval mint a fejlesztendő rendszerrel általában az alábbi alapvető követelményeket fektetjük le.

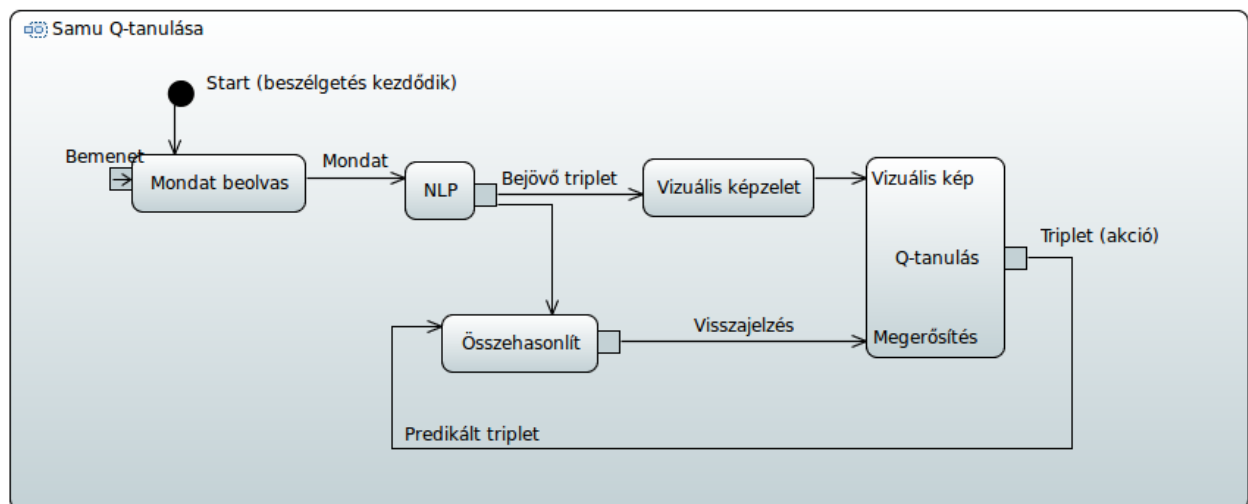
- Samu programja támogassa a különböző gépi tanulási algoritmusok kipróbálhatóságát.
- A különböző algoritmusokkal való kutatást/tesztelést egyfajta külső vezérlésnek tekinthetjük. Ez például akkor szembeűnő, ha összehasonlítjuk a prototípusok terminálon vagy hálózaton jött bemeneteinek feldolgozását az ezek hiányában beinduló korpuszok olvasásával. Mivel az előbbi kettő a Samu osztályban van implementálva, utóbbi a main függvényben. E külső vezérlés mellé, vagy méginkább e helyett vezessnk be egyfajta belső vezérlést, ahol Samu dönti el, hogy éppen mit csináljon.
- Samu minden kommunikációt mint korpuszokat mentsen le.

- Samu egyszerre egy beszélgetést tudjon folytatni (szemben azzal a távolabbi céllal, amikor a beszélgetéseket processzeknek feleltetjük meg és Samu egyfajta olyan operációs rendszer jellegű alakot öltön, aki időosztással tud több beszélgetést szimultán kezelni.)
- Samu legyen open source alapú, lehetőleg GNU GPL v3. A dokumentálás nyelve DocBook XML 5.1 szabvány.
- Samu tanító korpuszai legyenek open source alapú korpuszok.
- Samu fusson UNIX-típusú rendszereken, például GNU/Linux rendszereken.
- Samu a feldolgozott tanuló korpuszait tartsa perzisztens gyorsító-tárban.
- Samu használja ki a CUDA kártya adta extrém párhuzamosságot, ha a futtató környezet rendelkezik ilyen kártyával.

3.2.3. Szakterületi követelmények

3.2.3.1. Q-tanulás

A Samu tanulásában alapvető Q-tanuláshoz szükséges a megerősítéses tanulás alapjának, azaz a jutalomnak és a büntetésnek a biztosítása. Ezt Samu a [SAMU] kéziratban bemutatott módon valósítja meg, ezt foglalja össze a következő, az UML aktivitás diagramjainak jelöléseit használó ábra.



3.1. ábra. Samu Q-tanulása

3.2.3.2. A kutatási feladatok aspektus-orientált megfogalmazása

A fejlesztendő rendszer egyik alapeladata a kapcsolódó kutatás támogatása, az eddigi gyors protókat felhasználó kutatási tapasztalat azt sugallja, hogy a kutatás támogatását Samuba célszerű az aspektusorientált paradigma mentén átszövő vonatkozások formájában megvalósítani.

3.2.4. Felhasználói követelmények

A felhasználók felé Samu programja klasszikus csevegőprogramként jelenik meg.

- A csevegő felhasználók Samuból a szokásos csevegő programok megszokott felületét lássák.
- Samu be tudjon kapcsolódni internetes csevegőkbe. Ennek megfelelően ne csak 1:1, hanem 1:N résztvevős beszélgetéseket is tudjon kezelni.

- A csevegőknek legyen lehetőségük hitelesíteni magukat és így az azonosított felhasználók egyrészt nagyobb hatást tudjanak gyakorolni Samu tanulási folyamataira, másrészt róluk Samu legyen képes információkat tárolni, így emlékezni rájuk.
- A gondozó felhasználók tudják tanítani Samut.
- A Samut családi körben elérő gondozók meg tudják szakítani Samu tanulását vagy hálózati kommunikációját, tehát élvezzen prioritást a terminálon keresztüli elérés.

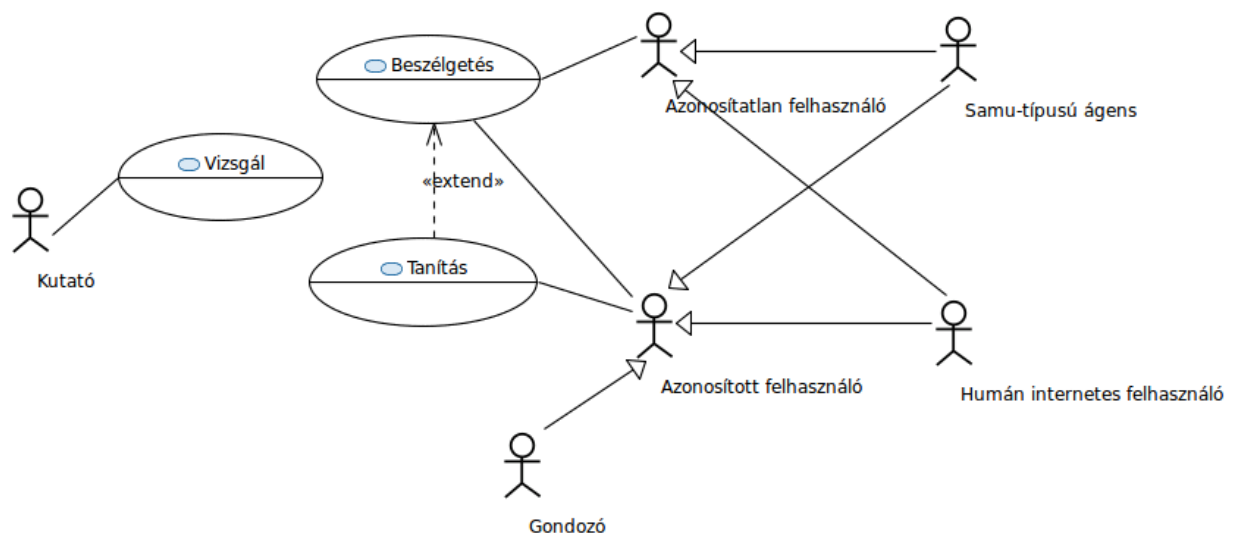
3.2.5. Rendszerkövetelmények

A követelmények feltárását az UML használati eset diagramjainak jelöléseivel készített ábrákkal segítjük.

3.2.5.1. Használati esetek

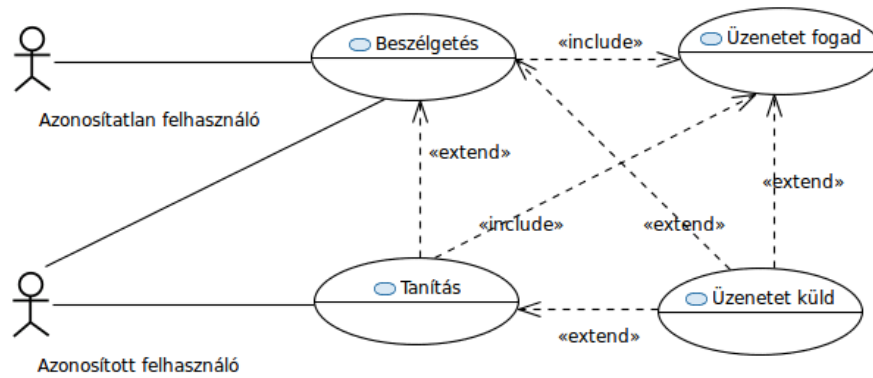
Azonosított vagy azonosítatlan felhasználó ugyanúgy lehet humán személy vagy Samu-típusú csevegő robot. Az azonosítottakra emlékszik Samu és ők tudják őt tanítani, az azonosítatlanok csak csevegni tudnak Samuval. Ez a megkülönböztetés azt jelenti, hogy az azonosítatlan csevegés nem tanítja Samut.

A kitüntetett gondozó felhasználó feladata Samu tanítása, amely alapja lehet egyoldalú beszélés Samuhoz, beszélgetés Samuval, vagy tanító korpuszok olvastatása Samuval. A kutató felhasználó speciális, Samunak olyan szolgáltatásokat kell felé nyújtania, amelyek Samu működését tudják monitorozni.



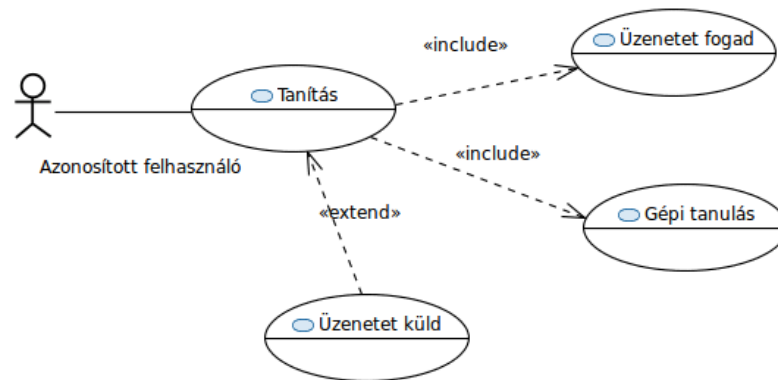
3.2. ábra. Samu használati esetei

A tanítás tehát nem csupán beszélgetés a gondozókkal, illetve a csevegés sem jelent szükségképpen minden bejövő mondatra választ.



3.3. ábra. A beszélgetés és a tanulás kapcsolata

Ugyanakkor a tanítás mindig üzenetfogadással és a megerősítéses tanulás alkalmazásával jár, a korábban vázolt **tanulási algoritmusnak** megfelelően.



3.4. ábra. A tanulás

3.2.6. Interfészek

TODO

Samu lelke sima szöveges állomány legyen, bz2-vel tömörítve.

3.3. Összefoglalás

TODO

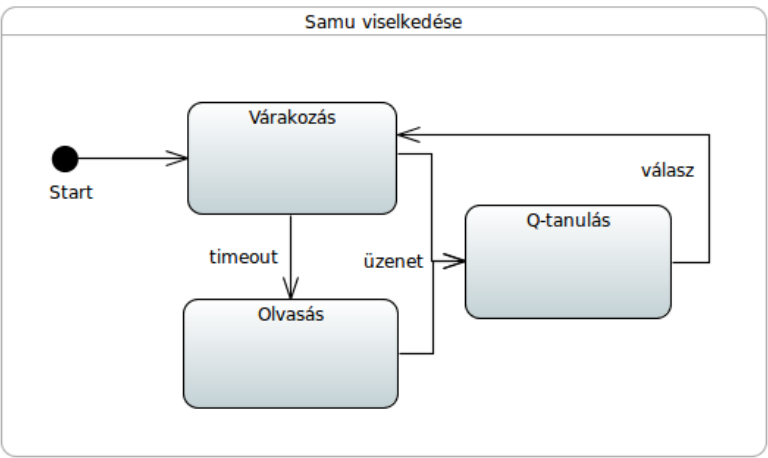
4. fejezet

Rendszermodellek

4.1. Rendszer-specifikáció

Jó gyakorlat a rendszerkövetelmények elemzésére az eldobható gyors prototípusok grafikus modelljeinek elkészítése. Ezek számos esetben (például az osztálymodell esetén osztálydiagramok formájában) automatikusan, reverse engineering jelleggel és egyben az agilis szemlélettel összhangban magukból az eldobható gyors prototípusok forráskódjaiból is generálhatóak.

4.1.1. Samu viselkedése madártávlatból



4.1. ábra. Samu állapotdiagramja

Állapot	Leírás
Várakozás	Samu várakozik a konzolról vagy a hálózatról érkező bejövő üzenetre (egy mondatra).
Olvasás	Samu az előkészített tanító korpuszokat olvassa (annak egy mondatát), azaz tanul ezekből a korpuszokból.
Q-tanulás	Samu a kapott mondattal végrehajtja a részben korábban már vázolt tanulási algoritmusát .

4.1. táblázat. Az állapotok leírása

Átmenet	Leírás
timeout	A megadott időhatáron belül sem a konzolról, sem a hálózatról nem érkezett üzenet.
üzenet	Üzenet (egy mondat) érkezett vagy a gondozói konzolról, vagy a hálózatról.
válasz	Samu a bejövő üzire válaszul visszahangozza a korábban már vázolt tanulási algoritmusának kimenetét.

4.2. táblázat. Az állapotátmenetek leírása

4.1.2. Samu adatmodelljei

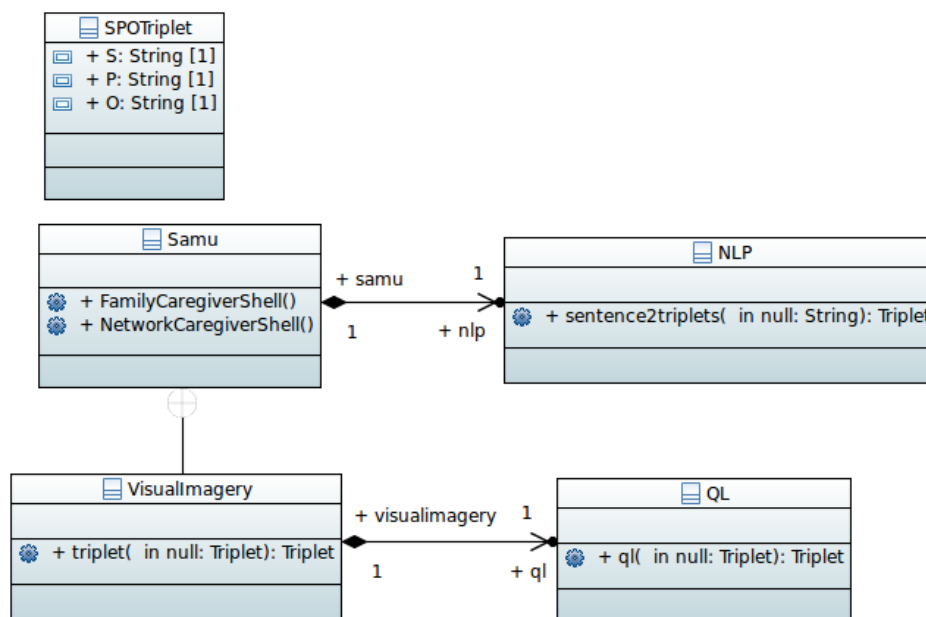
Samu jelenlegi eldobható gyors prototípusainak alapvető működése a természetes nyelvi mondatokból kibányászott Alany-Állítmány-Tárgy tripletekre épül, egy ilyen triplet alapesetben 3 szóból áll, a részletek tekintetében lásd a [SAMU] publikációt.

A tripletek kibányászására mint COTS komponenst a [opencog/link-grammar](https://github.com/robertostrom/openCogLinkGrammar) NLP eszközt használjuk.

Samu tanulása során minden triplethez egy egyedi MLP neurális hálózatot rendel, ezek perzisztens tárolásáról gondoskodni kell. A hálózatok mérete jellemző, a bemenő réteg neuronjainak a száma Samu vizuális képzeletének szélesség x magasság számával egyezik meg.

Az Amminadan prototípus bevezette, hogy a tripleteket egy LZW fában is tárolja Samu, lásd a részletek tekintetében a [SAMU] publikációt.

4.1.3. Samu objektummodellje



4.2. ábra. Samu osztálydiagramja

A Samu osztály reprezentálja Samu Nahshon verzióját. Samunak olyan integrált része a vizuális képzelete, hogy azt a Visual Imagery belső osztállyal absztraháltuk. Samuval az NLP-s COTS funkciókat elfedő NLP osztály, a vizuális képzelettel pedig Samu tanulását megvalósító QL osztály van kompozícióban.

5. fejezet

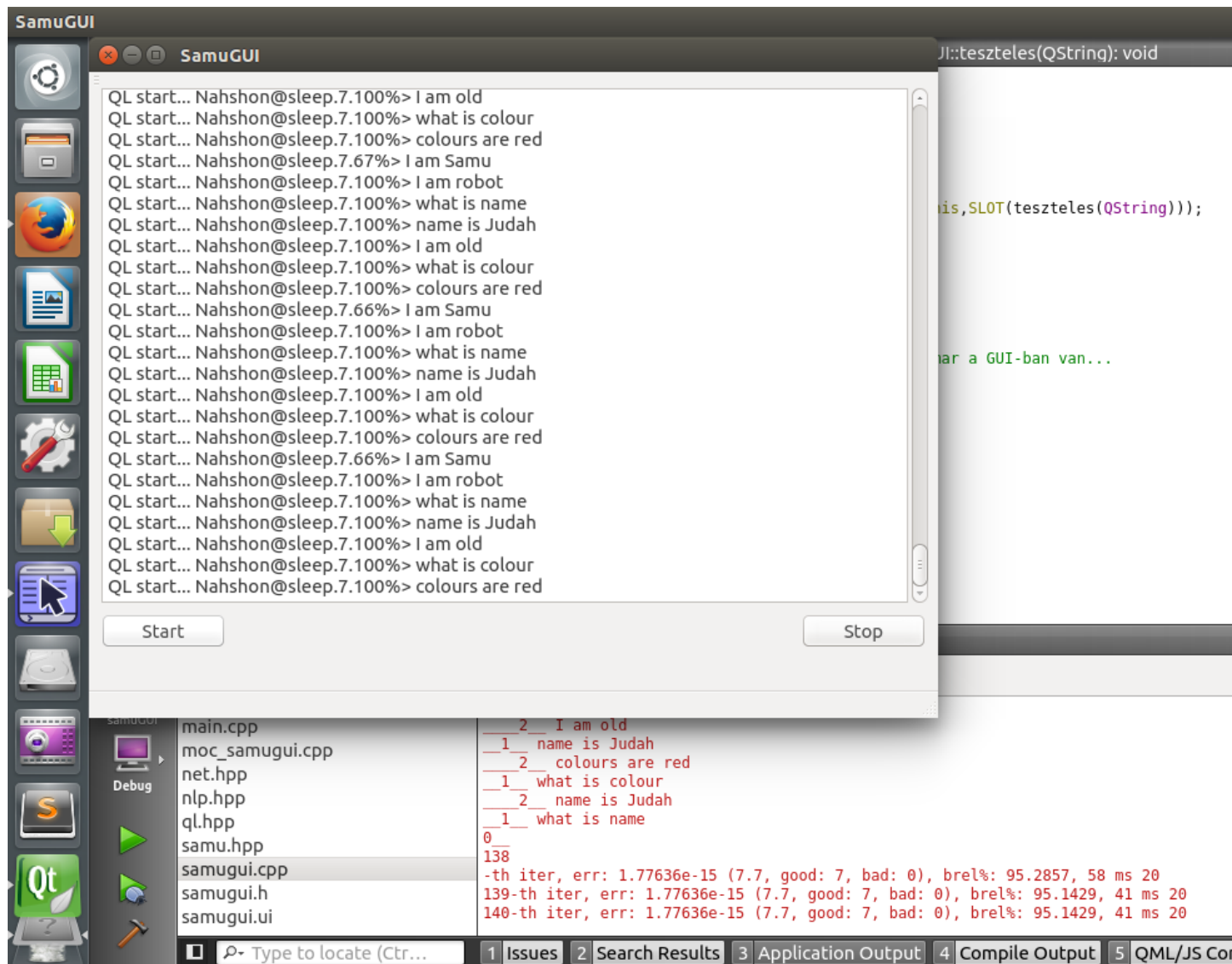
Samu GUI

5.1. Igény egy grafikus felhasználói felület készítésére

A grafikus felület készítésének ötlete a nehezen átlátható és kezelhető szöveges felület miatt merült fel. A 2015-ös prog2 kurzus harmadik védési feladatának indult (illetve jelen sorok írása időpontjában még csak indul, remélem meg is lesz a védés :)), viszont hamar kedvet kaptam a továbbfejlesztéshez, így remélem hosszú élet áll előtte.

Az elképzelés egy bárki által könnyen használható, sok kényelmi és alap funkciókkal ellátott felület, amit igény esetén a következő PTI-generációk könnyen bővíteni tudnak.

5.2. Kezelés, a felület kinézete



5.1. ábra. Samu GUI tanulás közben

A felület jellegének stílusát egyelőre a minimalista jelzővel illetném, ám a funkciók sora - remélem - az idők során bővülni fog.

A kezelés sem túl bonyolult; az ablak megjelenését követően a Start gombra kattintva Samu a háttérben elkezd tanulni, a fenti szövegmezőn láthatjuk a tanulás kimenetét. A stop gombra kattintva a tanulás leáll, Samu lelke elmentésre kerül.

Az ablak tanulás közbeni váratlan bezárása során a tanulás ugyancsak leáll és Samu lelke elmentésre kerül.

5.3. A fejlesztés lépései nagy vonalakban

A fejlesztés alatt Amminadab-ból (továbbiakban: Samu) indultam ki; ám az ő osztályaihoz jóformán nem nyúltam - kifejezetten a felület megírására fektettem hangsúlyt. A projekt során Amminadab működésében nem tapasztaltam változást.

A felület írása során a Qt függvényeit használtam fel. Cmake segítségével fordítottam, és itt adódott az első komolyabb probléma, miszerint egyszerűbb lett volna qmake-el fordítani az Qt osztályok miatt, ám így Samu megfelelő működését volt

nehéz biztosítani. Végül röpké félnapos keresgélés után arra jutottam, hogy a legmegfelelőbb lesz maradni a cmake-nél, és a CmakeLists.txt-ben hozzáadtam a Qt-s widgeteket. Így már a forráskódba is be tudtam illeszteni a megfelelő osztályokat.

Létrehoztam magát az ablakot, ezzel immár semmi probléma nem adódott.

Az ablak és Samu más külön szálon kell hogy fusson annak érdekében, hogy az ablak frissítése és Samu munkája ne függjön egymástól. Létrehoztam egy Thread osztályt, ennek a run() metódusába változtatás nélkül a main() metódusban szereplő kódot írtam.

A vezérlést slot-signal párok segítségével készítettem el, mindkét irányba. Tehát a gombok általi vezérlés és a kimenet küldése a grafikus felületre is ezekkel történt.

A felület alap funkciói így működőképessé váltak.

5.4. Készítette

2015. Debrecen, Mátó Attila Zoltán, PTI

6. fejezet

Rendszer-evolúció

6.1. Samu vezérlése

Már az Amminadab eldobható gyors protó végén, amikor itt Samu hálózati csevegő-szerver funkciója megvalósításra került, felmerült, hogy a következő lépésben kifejlesztendő csevegő kliens funkció az alábbi három használati esetet támogassa:

- Samu programja humán partnerrel beszélget (ez nem újdonság az eddigiekhez képest).
- Samu egy másik Samuval beszélget.
- Samu magában beszélget.

Az utóbbi két használati eset tudományos igényességű vizsgálata a [\[PRENATAL\]](#) kézirat kapcsán a jelen pillanatban is folyik. Ami a jelen dokumentum szempontjából érdekes az az, hogy el akarjuk kerülni az alábbi jellegű implementációt:

```
main()
{
    ...
    if(random() < a_magic_number)
        Samu_starts_talk();
    ...
}
```

hanem e helyett azt szeretnénk, hogy Samu „szabad akaratából” tegye ezt! Hasonló szituációt említünk meg a [\[SAMU\]](#) kéziratban:

„This is also reinforced by the fact that the incremental learning was implemented in the main.cpp and not in a class of Samu. From a point of view of a programmer, it means that the incremental approach is not yet really part of Samu.”

Tehát adott a kihívás, hogy megpróbáljuk mindezt, mint egyfajta belső vezérlést, Samu tanuló algoritmusába integrálni!

6.2. Beágyazott Samu

„Az ember legjobb barátja a koci” típusú rendszerek kapcsán már hangsúlyoztam [\[IMITGAME\]](#) annak fontosságát, hogy a fejlesztés mögött legyen izgalmas motiváció. Mint ilyet ebben a szekcióban potenciálisan összekapcsoljuk a Samu projektet a kutatásban már sikeresnek [\[OOCWC1\]](#), [\[OOCWC2\]](#), [\[OOCWC3\]](#), [\[OOCWC4\]](#) mondható OOCWC projekttel: ez a potenciális kapcsolat a beágyazott Samu.

A beágyazott Samu nem csupán egy „Az ember legjobb barátja a koci” típusú rendszer, ahol Samu az autókba tervezett MI része lehetne (mondjuk UI-ként), hanem az OOCWC alábbi konkrét fejlesztése, amelyet most speciálisan az OOCWC rendőrségi kiadására fogalmazunk meg.

Bevezetjük az OOCWC kliens ágensek olyan TCP-s parancskészlet kiegészítését, amely a meglévő (pl.: init, pos, car, route stb.) parancsok mellé olyan további protokoll-parancsokat ad, amelyek azt biztosítják, hogy a rendőr ágensek természetes nyelven (csevegés) kommunikálva tudják egymással összehangolni a tevékenységüket (intuitív értelemben asszociálhatunk a robotfocisay parancsára).

Tekintsünk egy konkrét használati eset csipetet az ötlet megvilágítására: tegyük fel, hogy ezt mondja a rendőr multiágenseket összefogó diszpécser ágens:

- remek fánkot vett a 7. egység a Kassai úton
- gyanús elemek mentek be az ékszerboltba a Füredin

nyilván azok az ágensek, akik „értik” ezeknek a forgalmazott üzeneteknek a tartalmát és ennek megfelelően a Füredire mennek várhatóan előnybe kerülnek az elfogások számában.

Ez természetesen egy új kiadást is kíván, mert erre az átszövő vonatkozásra a szerveroldalt is fel kell készíteni például. Ez az új kiadás lesz a *SAMU-OOCWC*.

7. fejezet

Irodalomjegyzék

7.1. Kutatási cikkek

- [COP] Bátfai, Norbert, *Conscious Machines and Consciousness Oriented Programming*, <http://arxiv.org/abs/1108.2865> , 2011.
- [IMITGAME] Bátfai, Norbert, *Turing's Imitation Game has been Improved*, <http://arxiv.org/abs/1509.00584> , 2015.
- [OOCWC1] Bátfai, Norbert, Besenczi, Renátó, Mamenyák, András, Ács Ispány, Márton, *OOCWC: The Robocar World Championship Initiative*, 2015 IEEE 13th International Conference in Telecommunications (ConTEL) <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7231223> , 2015.
IEEE, 10.1109/ConTEL.2015.7231223, 1-6, 2015.
- [OOCWC2] Bátfai, Norbert, Besenczi, Renátó, Mamenyák, András, Ács Ispány, Márton, *Traffic Simulation based on the Robocar World Championship Initiative*, 2015.
Infocommunications Journal, 7:(3), 50-59, 2015.
- [OOCWC3] Besenczi, Renátó, Szilágyi, Mihály, Bátfai, Norbert, Mamenyák, András, Oniga, István, Ács Ispány, Márton, *Using Crowdsensed Information for Traffic Simulation in the Robocar World Championship Framework*, 2015.
Cognitive Infocommunications (CogInfoCom), 2015 6th IEEE Conference on, 333-337, 2015.
- [OOCWC4] Besenczi, Renátó, Katona, Tamás, Ács Szilágyi, Mihály, *A Fork Implementation of the Police Edition of the OOCWC System*, 2015.
Cognitive Infocommunications (CogInfoCom), 2015 6th IEEE Conference on, 163-164, 2015.
- [PRENATAL] Bátfai, Norbert, *Samu in his prenatal development*, (kézirat, írása folyamatban) , 2015.
- [SAMU] Bátfai, Norbert, *A disembodied developmental robotic agent called Samu Bátfai*, <http://arxiv.org/abs/1511.02889> , 2015.
-

8. fejezet

Tárgymutató

C

csevegő robot, [3](#)

CUDA, [6](#)

D

developmental robotic, [3](#)

DocBook, [6](#)

XML, [1](#)

E

eldobható gyors prototípus, [3](#), [9](#), [10](#), [14](#)

G

GNU

GPL, [6](#)

Linux, *lásd* Linux

L

Loebner díj, [3](#)

M

multiágens, [15](#)

O

OOCWC, [14](#), [15](#)

open source, [6](#)

Q

Q-tanulás, [3–6](#)

S

SAMU-OOCWC, [15](#)

SPO, [4](#)

T

Turing teszt, [3](#)

U

UML, [5–7](#)

UNIX, [6](#)
