BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2017
http://excel.fit.vutbr.cz

Nástroj pre skriptovanie scenárov evaluácie leteckého kokpitu

Richard Granec*

Abstrakt
Cieľom tejto práce je vytvorenie nástroja pre správu scenárov bežiacich na pozadí leteckého simulátora. Úlohou týchto scenárov je možnosť prispôsobiť prostredie leteckého simulátora a v konečnom dôsledku aj celý let pre jeho kontrétne využitie najmä pri testovaní zručností pilota alebo nového technického vybavenia. Riešenie pozostáva z komunikačnej aplikácie, ktorá komunikuje so simulátorom a získané dáta odosiela na NoSQL server, a správcu scenárov, ktorý zvolený scenár spustí a zaznamenáva jeho činnosť. Pre výber konkrétneho scenára a jeho následnú kontrolu slúži webové grafické rozhranie. Takto vytvorený softvér bol testovaný a následne bude využívaný aj leteckým odvetvím firmy Honeywell.

Kľúčové slová: Správca scenárov — Letecký simulátor — NoSQL — Redis — Scenáre riadiace let

Priložené materiály: Demonstration Video

*xgrane00@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Úvod

Letecké simulátory sú často využívané nie len pre zábavu, ale v praxi sú nimi testované znalosti a zručnosti pilota, prípadne technického vybavenia leteckých dopravných prostriedkov. V týchto prípadoch použitia je mnohokrát požadovaná možnosť riadenia behu simulátora externými procesmi, čím je možné prispôsobiť prostredie simulátora a v konečnom dôsledku aj celý let konkrétnemu testovanému subjektu. Počas takto vykonávaných testov vzniká pre možnosť následnej analýzy a vyhodnotenia vykonávaných testov potreba zaznamenávať dáta a konfiguráciu technického vybavenia, priebeh letu a činnosť pilota. Všetky takto zadefinované požiadavky je možné napĺňať využitím scenárov, ktoré bežia na pozadí leteckého simulátora a komunikujú s ním. Táto práca vznikla ako reakcia na potrebu vytvorenia správcu vyššie opísaných scenárov, ktorý by umožnil výber požadovaného scenára, zaistil by spoľahlivú komunikáciu scenárov a simulátora a následne by vhodnou metódou zobrazoval priebeh a stav, v ktorom sa spustený scenár nachádza. Pre možnosť výberu požadovaného scenára a sledovania jeho priebehu je využité webové grafické rozhranie, vďaka ktorému môžeme obsluhovať vytvorenú aplikáciu z rôznych zariadení. Návrh na vypracovanie tejto témy, následné testovanie zhotovenej aplikácie a taktiež aj jej použitie, prebieha v spolupráci s leteckým odvetvím firmy Honeywell.

Kapitola číslo dva opisuje technológie, využívané vytvorenou aplikáciou. V tretej kapitole sa nachádza štruktúra výslednej aplikácie. Štvrtá kapitola opisuje funkcionalitu troch základných častí systému.

2. Evaluácia leteckého kokpitu

2.1 Scenáre riadiace a zaznamenajúce let

Letecké simulátory *Microsoft flight simulator* a *XPlane* poskytujú svojim užívateľom možnosť výberu dopravného prostriedku, štartovacieho letiska, prípadne úpravy času letu a poveternostných podmienok, avšak po takto spustenom lete už ho nie je možné ovplyvniť, respektíve prispôsobiť. Pri využívaní týchto simulátorov pre potreby testovania zručností pilota alebo nového elektronického vybavenia, ktoré podlieha testovaniu z dôvodu jeho možného využívania v kokpite lietadla, je často potrebné vedieť ovplyvniť a zmeniť spustenú simuláciu. Vyžadované sú často úpravy stavu paliva, pozastavenie simulácie, úprava stavu určitých komponentov, napríklad odstavenie motorov lietadla pri testovaní krízových situácií a mnoho ďalších. Práve scenáre nám prinášajú možnosť ako externými procesmi ovplyvniť chod simulácie a prispôsobiť ju tak požadovaným podmienkam. Po úspešnej konfigurácií letu a počas celej doby simulácie vzniká pre možnosť následnej analýzy letu, správania sa a reakcií pilota potreba zaznamenávania priebehu letu, stavu ovládacích prvkov dopravného prostriedku a konfigurácií sledovaného technického vybavenia. Tieto potreby nám opäť umožňujú napĺňať scenáre, ktoré okrem riadenia letu umožňujú zaznamenávať všetky potrebné premenné a ukladať ich do zvolených logovacích súborov.

Scenáre nám predstavujú programy spustené na pozadí leteckého simulátora, ktoré s ním komunikujú. Môžu byť implementované pomocou rozličných programovacích jazykov, ale podmienkou je podpora daného jazyka doplnkom simulátora, teda existencia komunikačnej knižnice.

2.2 NoSQL databáza

NoSQL systémy sú taktiež často nazývané „Not only SQL“ pre poukázanie podpory dotazovacích jazykov podobným SQL. Poskytujú mechanizmy pre uloženie a načítanie dát, ktoré sú modelované spôsobom odlišným od relačných databáz. Podporujú nerelačný dátový model a distribuovanú architektúru. Sú často používané pre spracovanie veľkého množstva dát a pri webových aplikáciách pracujúcich v reálnom čase. Základné vlastnosti týchto databáz sú: jednoduchosť dizajnu, nenáročné horizontálne skálovanie clustermi počítačov (čo je problém pri relačných databázach) a lepšia kontrolu nad dostupnosťou. Dátové štruktúry využívané týmito databázami sú predovšetkým typu *kľúč-hodnota*, *graf*, *wide column* a *document store*. Práve tieto typy umožňujú vykonávať mnohé operácie nad uloženými dátami rýchlejšie, než tomu tak je v relačných databázach [1, 2].

2.3 Redis databáza

Redis [3] je nerelačná databáza, často opisovaná ako pamäťovo perzistentné uloženie typu kľúč-hodnota, avšak nejedná sa o presnú definíciu, keďže Redis je oveľa viac, než len jednoduché uloženie typu kľúč-hodnota. Realita je taká, že Redis ponúka oficiálne päť dátových štruktúr:

- string,
- hash,
- lists,
- sets,
- sorted sets.

Využíva rovnaký koncept databázy, ktorý je nám už známy. Databáza obsahuje sadu dát. Typické využitie databázy je zoskupenie všetkých dát aplikácie dokopy a ponechanie ich separátne od ostatných aplikácií. Jednotlivé databázy sú identifikovateľné pomocou ich indexu.

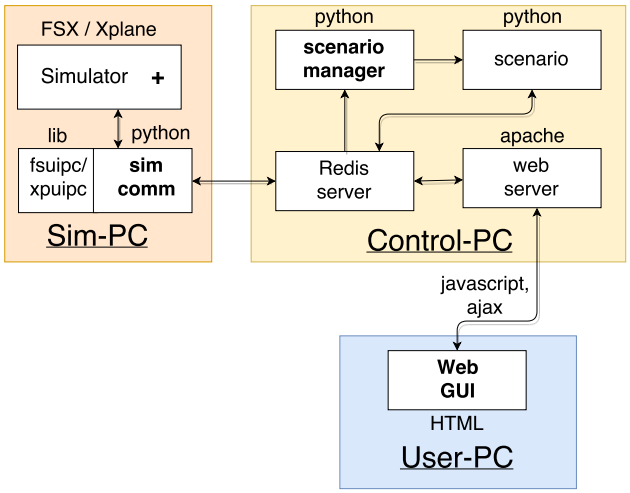
Kým Redis je viac než uložisko typu kľúč – hodnota, v jadre obsahuje každá z dátových štruktúr prinajmenšom kľúč a k nemu prinaležiacu hodnotu.Kľúče tvoria spôsob, ktorým sú dáta identifikované a môžu mať viacero podôb. Ako príklad si môžeme uviesť *users:forest*. Dá sa očakávať, že tento kľúč bude obsahovať informácie o užívateľovi *forest*. Dvojbodka v tomto prípade nehrá žiadnu rolu a je to len spôsob, akým si užívateľ môže organizovať svoje kľúče.

Hodnoty reprezentujú aktuálne dáta priradené ku zvolenému kľúču. Niekedy je vhodné uložiť dané dáta ako reťazce znakov, niekedy ako čísla a inokedy je vhodné uložiť ich ako serializované objekty typu *JSON*, *XML*, prípadne iné formáty. Vo väčšine prípadov Redis zaobchádza s hodnotami ako s poľom bytov a nezaujíma sa o to, čo sa v nich nachádza [4].

3. Nástroj pre správu scenárov

3.1 Štruktúra výslednej aplikácie

Medzi hlavné požiadavky výslednej aplikácie patrí snaha, o čo najnižšie zaťaženie počítača, na ktorom je letecký simulátor spustený. Tento simulačný počítač je pomenovaný *Sim-PC*. Splnenie tejto požiadavky je podstatné najmä z dôvodu, že oba nami využívané letecké simulátory vyžadujú pre svoj plynulý chod a čo najreálnejšie grafické zobrazenie vysoké požiadavky technického vybavenia počítača, na ktorom budú spustené. Práve preto simulačný počítač obsahuje len jednoduchú komunikačnú aplikáciu, ktorá opakovane číta požadované premenné simulátora a odosiela ich na server Redis. Ten sa nachádza na kontrolnom počítači, označený ako *Control-PC*. Táto časť systému je taktiež zodpovedná za správu scenárov. Scenáre,

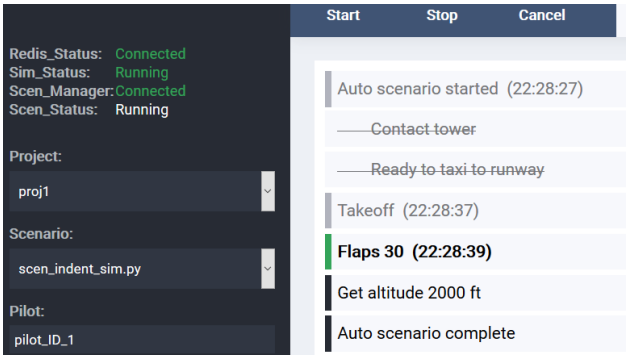


Obrázok 1. Schéma výslednej aplikácie

ktoré sú spravidla na tomto zariadení aj uložené, sú sprístupnené pre webový server a po výbere požadovaného scenára sú následne správcom scenárov aj spustené. Posledným prvkom tejto časti systému je aj samotný webový server, pomocou ktorého sú užívateľovi zobrazované a sprístupnené potrebné dáta, a ovládacie prvky aplikácie. Schéma výslednej štruktúry bola navrhnutá pre optimálny chod celého systému a jej výsledná podoba je zobrazená na Obrázku 1. Nakoľko je aplikácia rozdelená do troch častí, teda troch samostatných počítačov, toto rozdelenie nie je nutné dodržať a jej úplná funkcionalita je dosiahnuteľná aj s využitím len jedného počítača.

3.2 Webové užívateľské rozhranie

Pre interakciu s vytvorenou aplikáciou sme zvolili webové užívateľské rozhranie, z dôvodu možnosti jeho zobrazenia a obsluhy na rôznych vzdialených zariadeniach. Medzi základné požiadavky patrí možnosť výberu projektu, následne odpovedajúcemu scenáru daného projektu a taktiež ID pilota. Následne vzniká potreba možnosti spustenia zvoleného scenára, jeho ukončenie, prípadne pozastavenie. Ďalšou doplnkovou funkciou je aj zobrazenie aktuálneho stavu komunikačnej aplikácie, správcu scenárov, pripojenia na server Redis a spustenia simulátora. Túto základnú funkcionalitu dopĺňa vizualizácia krokov spusteného scenára, kde sú užívateľovi zobrazené už vykonané kroky (sivá farba), kroky ktoré boli preskočené, teda nevykonané (sivá farba, prečiarknuté), aktuálny krok (zelené návestie) a budúce kroky (čierne návestie). Opísané prvky môžeme vidieť na Obrázku 2. Možnosť zobrazenia nasledujúcich krokov máme vďaka predspracovaniu a analýzy scenára tesne pred jeho spustením. Vizualizovaná je taktiež aj hĺbka zanorenia príslušného kroku v podmienkových blokoch. Počas vykonávania scenárov a zobrazenia daných krokov sú užívateľovi poskytnuté



Obrázok 2. Základné prvky užívateľského rozhrania

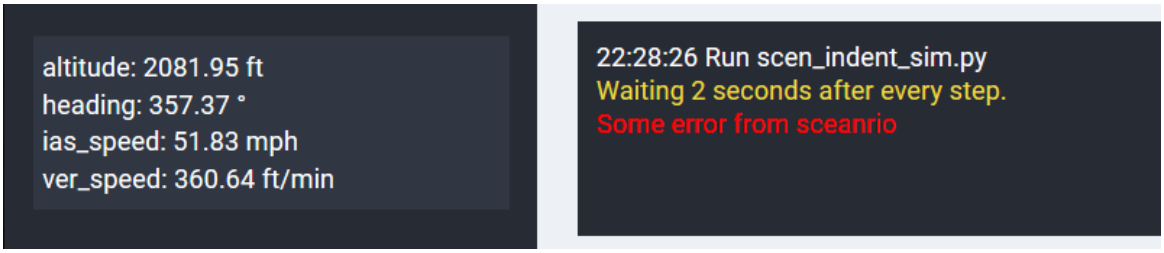
aj dodatočné informácie o používanom dopravnom prostriedku, akými sú jeho aktuálna výška, kurz, indikovaná vzdušná rýchlosť a vertikálna rýchlosť. Táto časť grafického rozhrania sa nachádza vľavo, v spodnej časti užívateľského rozhrania, a môžeme ju vidieť na Obrázku 3. V pravej časti je možné vidieť časť konzoly, ktorá dopĺňa informácie o spustených scenároch, ich chybách a dôvodoch ukončenia. K dispozícií sú tri typy správ, ktoré je možné konzolou zobraziť, a to informačná, výstražná a chybová.

3.3 Komunikácia a sprístupnenie dát

Najpodstatnejším prvkom komunikácie je server Redis. Zaisťuje prepojenie jednotlivých častí systému a umožňuje spoľahlivú a rýchlu komunikáciu aj v prípade, že sa jednotlivé prvky nachádzajú na rozličných počítačov.

Všetká komunikácia so simulátorom a nami spustenou aplikáciou prebieha cez verejne dostupný doplnok *fsuipc* [5] prípadne *xpuipc* [6]. Tento doplnok je doinštalovaný do príslušného simulátora a spolu s priloženou komunikačnou knižnicou nám umožňuje, aby spustený simulátor komunikoval s rôznymi externými zdrojmi. Touto cestou sú získavané potrebné dáta, ktoré sú pomocou nami vytvorenej komunikačnej aplikácie odoslané na server Redis. Táto činnosť zabezpečuje možnosť analýzy prebiehajúceho letu a jeho logovanie.

Informácie o priebehu letu sú najčastejšie využívané spusteným scenárom, ktorý je pripojený na server Redis a väčšinou periodicky číta dáta. Použité môžu byť scenáre, ktoré dáta len čítajú a ukladajú ich pre potreby neskoršej analýzy, avšak môžu byť využité scenáre, ktoré práve dáta na server iba zapisujú. Po tom, čo sú dáta zapísané do databázy, komunikačná aplikácia bežiaca na pozadí simulátora, označená na Obrázku 1 ako *sim_comm*, zaregistruje túto zmenu a zapíše dané dáta do simulátora. Takéto scenáre môžu byť použité najmä po spustení simulátora pred začatím letu, teda testu, a slúžia na upresnenie konfigurácie daného leteckého prostriedku, ako napríklad úprava stavu paliva, ovládacích prvkov a im podobné.



Obrázok 3. Vľavo informácie o stave lietadla, vpravo časť informačnej konzoly

Ďalšou časťou systému, s ktorou je nutné zabezpečiť komunikáciu, je webový server. Ten využíva čítanie dát z databázy, pre ich vizualizáciu a informovanie užívateľa, v akej aktuálnej výške sa lietadlo nachádza, kam smeruje, a tiež získava informácie o kroku scenára, v ktorom sa aktuálne spustený test nachádza. Posledným komunikujúcim prvkom je správca scenárov označený na Obrázku 1 *scenario_manager*. Ten využíva toto pripojenie na získanie informácií o zvolenom scenári, pilotovi a aktuálnom stave scenára, čiže jeho spustenie, prípadne ukončenie.

4. Aplikácia pre sprístupnenie dát a riadenie letu

Výsledná implementácia je tvorená z troch základných častí, ktoré sú opísané v tejto kapitole.

4.1 Správca scenárov

Hlavnou úlohou správcu scenárov je zistenie, ktorý scenár bol užívateľom zvolený a jeho následné spustenie. Nasleduje kontrola v ktorej časti, respektíve kroku, sa aktuálne nachádza. Pre implementáciu tohto správcu bol zvolený programovací jazyk *python* vo verzii *3.6*, ktorý využíva pre komunikáciu s databázou doplnok jazyka *python* s názvom *redis-py* [7].

Pri spustení správcu prebieha najskôr kontrola dostupnosti serveru Redis a v prípade jeho nedostupnosti skript na neho čaká. Po úspešnom spustení sa dostáva do stavu, kedy je očakávaný vstup od užívateľa, a to zvolením príslušného projektu, scenára a ID pilota vo webovom grafickom rozhraní. Po zadaní požadovaného scenára je vybraný scenár spustený správcom, ako jeho podproces. Ako môžeme vidieť na Obrázku 3, webové rozhranie disponuje aj konzolou. Po spustení scenára sú do tejto konzoly správcom odoslané správy, informujúce užívateľa o názve a čase spustenia scenára. V prípade, že scenár je chybný a nepodarí sa ho spustiť, alebo je počas jeho vykonávania nečakane ukončený, úlohou správcu je zachytiť túto chybu a upozorniť na ňu užívateľa odoslaním dôvodu ukončenia scenára.V prípade úspešného vykonania scenára, správca opäť informuje užívateľa o úspešnom prevedení a pripraví sa, teda čaká, na spustenie ďalšieho scenára.

4.2 Komunikačná aplikácia

Cieľom tejto časti aplikácie je nadviazať a udržať spojenie so spusteným simulátorom, za účelom výmeny dát. Využívané sú verejne dostupné doplnky použitých leteckých simulátorov a k nim prislúchajúce komunikačné knižnice, umožňujúce nami vytvorenej komunikačnej aplikácie spoľahlivé a rýchle čítanie dát zo simulátora, prípadne ich zápis. Všetky dáta, ku ktorým nám simulátor v spolupráci s doplnkom umožňuje pristúpiť, sú uložené v špeciálnom bloku pamäte. Ku daným premenným je možné pristúpiť pomocou konkrétnych offsetov, ktoré nám udávajú presnú polohu danej premennej v pamäti. Zvolené umiestenie a veľkosť čítaných, prípadne zapisovaných dát, tvoria parametre funkcie knižnice, dodávanej k danému doplnku simulátora, a umožňujú nám prácu s požadovanými dátami. Keďže komunikačná aplikácia je implementovaná taktiež v jazyku *python*, na komunikáciu je využitá príslušná knižnica *pyuipc.pyd*. Táto knižnica nahradila využitie knižnice v jazyku *C*, ktorú sme si zvolili na počiatku projektu. K výmene došlo pri rozdeľovaní jednotlivých častí systému, a to z dôvodu jednoduchšieho a spoľahlivejšieho priameho pripojenia na server Redis.

4.3 Webový server

Webový server tvorí jadro grafického rozhrania a s jeho využitím sú užívateľovi zobrazované potrebné informácie a poskytnuté vhodné prvky na obsluhu aplikácie. Pre korektné a prehľadné zobrazenie sú využité technológie *HTML* a *CSS*. Na strane servera je využívaná technológia *PHP*, ktorá slúži na získanie a zápis dát do našej databázy. Taktiež je využívaná aj na prehliadanie lokálnej zložky, v ktorej sú uložené jednotlivé projekty a nim prislúchajúce scenáre. Vďaka tomu sú užívateľovi zobrazené a na výber ponúknuté projekty, zodpovedajúce hierarchií ich uloženia v danom počítači. Viacero jednotlivých prvkov grafického rozhrania sa počas chodu celého systému pravidelne mení. Pre ich aktuálne zobrazenie využíva webový server technológie *JavaScript* a *Ajax*. Technológia *Ajax* nám umožňuje dynamicky meniť obsah zobrazovanej stránky, bez potreby jej kompletného znovunačítania. Využitie týchto technológií nám umožňuje

tvorbu užívateľského rozhrania, ktoré užívateľovi poskytuje vždy aktuálne informácie o stave simulátora, letu a spusteného scenára s minimálnym oneskorením.

4.4 Spôsob konfigurácie aplikácie

Obe základné časti systému, je možné prispôsobiť pomocou konfiguračného súboru. Ten obsahuje absolútne cesty k príslušným projektom a scenárom, adresu servera Redis, no pomocou neho sú užívateľom volené aj dáta, ktoré majú byť sprístupnené komunikačnou aplikáciou. Uvedená je adresa, respektíve offset požadovanej premennej, jej veľkosť a taktiež pomenovanie danej premennej, ktoré si užívateľ volí a priradeným menom môže následne v scenári k danej hodnote pristúpiť.

5. Záver

Výsledkom tejto práce je aplikácia, pomocou ktorej je možné spustiť príslušný scenár letu a sledovať proces jeho vykonávania. Zvolený scenár nám umožňuje upraviť priebeh letu a konfiguráciu použitého leteckého prostriedku, prípadne zaznamenávať priebeh spustenej simulácie. Základnú funkcionalitu dopĺňa informačná konzola, prednačítanie všetkých krokov zvoleného scenára a ich vizualizácia rozlišujúca preskočené, vykonané a budúce kroky.

V práci sú opísané použité technológie ako *Ajax* a *NoSQL* databázy, pomocou ktorých bolo možné zrealizovať daný projekt. S navrhnutým rozložením príslušných častí systému a využitím kontrolného počítača, označovaného *Control-PC*, nedochádza ku nadbytočnému zaťaženiu počítača, na ktorom je spustený simulátor, čo taktiež patrí medzi požiadavky na výslednú aplikáciu. Na základe týchto vlastností vytvorenej aplikácie môže užívateľ spustiť požadovaný scenár, a tak jednoduchým spôsobom upraviť priebeh letu, prípadne jeho priebeh zaznamenať.

Výsledný systém bol počas svojho vývoja testovaný a následne aj nasadený vo firme Honeywell. Možnosť rozšírenia vytvorenej aplikácie sa odvíja aj od jej dlhodobejšieho používania v praxi. V aktuálnej podobe prichádza do úvahy pridanie možnosti spustenia viacerých scenárov naraz a ich prehľadná, separátna vizualizácia. Vhodným rozšírením je aj možnosť úpravy konfiguračného súbora a uloženie nového scenára pomocou webového rozhrania.

Poďakovanie

Moje poďakovanie patrí predovšetkým vedúcemu práce Prof. Ing. Adamovi Heroutovi, Ph.D., za jeho odborné rady a motiváciu.

Literatúra

[1] Marek Rychlý and Dušan Kolár. NoSQL databáze. http://www.fit.vutbr.cz/~rychly/public/docs/slides-nosql-databases/slides-nosql-databases.pdf. [Online; navštíveno 21.03.2017].

[2] Irena Holubová, Jiří Kosek, Karel Minařík, and David Novák. *Big Data a NoSQL databáze*. Grada, 2015. ISBN: 9788024754666.

[3] Karl Seguin. The little redis book. http://openmymind.net/redis.pdf. [Online; navštíveno 21.03.2017].

[4] Eric Redmond and Jim Wilson. *Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement*. The Pragmatic Programmers, 2013. ISBN: 9781934356920.

[5] Peter Dowson. FSUIPC4, plugin MS flight simulator x. http://www.schiratti.com/dowson.html. [Online; navštíveno 05.04.2017].

[6] XPUIPC, plugin XPlane. http://www.tosi-online.de/XPUIPC/What_is_it.html. [Online; navštíveno 05.04.2017].

[7] Andy McCurdy. redis-py. https://github.com/andymccurdy/redis-py. [Online; navštíveno 05.04.2017].