

REZUMAT:

Documentație de proiectare tehnici de transfer automat, adaptiv și sensibil la context între dispozitive smart și IVIS

Proiect complex:

PN-III-P1-1.2-PCCDI-2017-0917

Proiect component:

P2 - Comunicații eficiente bazate pe dispozitive inteligente în scenarii interactive de realitate augmentată pentru autovehicule

Parteneri:

Universitatea Ovidius din Constanța

Universitatea Ștefan cel Mare din Suceava

Autori:

Universitatea Ovidius din Constanța

Prof.univ.dr. Dorin-Mircea POPOVICI
Conf.univ.dr. Dragoș-Florin SBURLAN
Conf.univ.dr. Crenguța Mădălina PUCHIANU
Lect.univ.dr. Elena BĂUTU
Asist.cerc.drd. Emanuela BRAN

Universitatea Ștefan cel Mare din Suceava

Prof.dr.ing. Radu-Daniel VATAVU
Prof.dr.ing. Ștefan-Gheorghe PENTIUC
Conf. univ. dr.ing. Ovidiu-Andrei SCHIPOR
Asist.cerc.drd. Laura-Bianca BILIUS

Cuprins

Cuprins	2
1. Introducere	3
2. Filtru configurabil pentru notificările livrate de telefoane smart	3
3. Aplicație pentru transferul automat a datelor între dispozitive smart personale și sistemul rulând în interiorul autovehiculului	5
4. Modelarea conceptuală a mesajelor și asigurarea persistenței informațiilor în interiorul autovehiculului	5
6. Referințe	7

Versiunea extinsă și completă a acestui document a fost înaintată UEFISCDI prin platforma EVOC. Aceasta versiune completă va fi publicată online pe site-ul proiectului după valorificarea rezultatelor cercetării printr-o publicație științifică!

© Universitatea Ovidius Constanța
© Universitatea Ștefan cel Mare din Suceava

Reproducerea sau utilizarea integrală sau parțială a prezentului document în orice publicații și prin orice procedeu (electronic, mecanic, fotocopiare, multiplicare etc.) este interzisă dacă nu există acordul scris al partenerilor (Universitatea Ovidius din Constanța și Universitatea Ștefan cel Mare din Suceava).

1. Introducere

Etapa nr. 2 aferentă anului 2019 de implementare a proiectului component P2 a constat în realizarea activității 2.3 a proiectului complex „Implementare de noi tehnici pentru transferul de conținut digital între dispozitive smart și sisteme in-vehicle” de tip A2, cercetare industrială.

Activitățile tehnice și științifice întreprinse în cadrul etapei anului 2019 au constat în:

- Proiectare de tehnici de transfer manual a conținutului digital între dispozitive smart și sistemul rulând în cadrul autovehiculului (denumit IVIS).
- Proiectare de tehnici de transfer automat, adaptiv și sensibil la context între dispozitive smart și sistemul rulând în cadrul autovehiculului IVIS.
- Proiectare de tehnici de realitate augmentată pentru vizualizarea transferului digital și implementarea software a tehnicilor de transfer.
- Evaluarea performanței tehnice și redactarea raportului de testare.
- Pregătirea, redactarea, trimiterea în vederea evaluării și publicării de lucrări științifice în tematica proiectului component P2.

Prezentul document reprezintă rezumatul livrabilului nr. 2/2019 intitulat “Documentație de proiectare tehnici de transfer manual, adaptiv și sensibil la context a conținutului digital între dispozitive smart și IVIS”.

În cadrul etapei curente, am continuat dezvoltarea componentelor software de tip producători și consumatori de evenimente și mesaje pentru arhitectura Euphoria [Schipor *et al.*, 2019] în vederea adăugării de noi funcționalități privind adaptabilitatea la condițiile de trafic, astfel:

- Proiectarea unui **filtru configurabil pentru notificările** livrate de către telefonul smart al conducătorului auto (prezentat în secțiunea 2 a acestui raport).
- **Demonstrarea adaptării notificărilor** livrate de dispozitivele smart la condițiile de trafic folosind senzorii integrați (prezentată în secțiunea 2).
- **Demonstrarea determinării contextului** în care este condus autovehiculul folosind o componentă software specializată (prezentată în secțiunea 2).
- **Demonstrarea transferul automat a conținutului digital** între dispozitivele smart (*e.g.* ceas smart) ale conducătorului auto și sistemul auto (secțiunea 3).

2. Filtru configurabil pentru notificările livrate de telefoane smart

Am proiectat și implementat un filtru care permite conducătorului auto să configureze programul de livrare a notificărilor redade de către telefonul smart, după cum urmează: (a)

notificări pe care conducătorul auto este dispus să le primească în condiții *normale* de condus, respectiv (b) notificări pe care este dispus să le primească în condiții de condus *riscante*. În timpul funcționării, aplicația care implementează filtrul analizează tipul notificărilor în conformitate cu configurația specificată de către utilizator (*adaptabilitate la utilizator*), dar și în funcție de condițiile de condus (*adaptabilitate la contextul utilizării*), determinate automat folosind GPS-ul și accelerometrul integrat în telefonul smart care implementează filtrul.

În vederea includerii acestor funcționalități, am realizat o actualizare a arhitecturii software a sistemului dezvoltat în cadrul etapei anterioare a proiectului folosind Euphoria [Schipor *et al.*, 2019] prin analiza, proiectarea, implementarea și testarea noilor cerințe de adaptabilitate. Astfel, în faza de analiză a cerințelor, am identificat noi actori software, cum ar fi accelerometrul încorporat în telefonul smart. Când telefonul smart este plasat pe bordul autovehiculului, accelerometrul furnizează date care, în urma procesării, pot oferi informații privind apariția diverselor evenimente, cum ar fi autovehiculul aflat în repaus, în mișcare rectilinie sau curbilinie. În această situație, sistemul din interiorul autovehiculului va schimba automat și adaptiv opțiunile de filtrare a diverselor notificări ale telefonului. De asemenea, am identificat noi cazuri de utilizare.

Fiecare caz de utilizare a fost analizat atât *structural* prin crearea unei diagrame UML¹ de activități, cât și *comportamental* prin crearea unei diagrame UML de secvență de sistem [UML]. În continuare, fiecare caz de utilizare a fost proiectat prin crearea unei diagrame UML de secvență, respectiv de clase.

În vederea obținerii unei perspective mai largi asupra contextului în care se află conducătorul auto, am dezvoltat o componentă software denumită *OBDDII_Producer_VO* pentru achiziția informațiilor de bord ale autovehiculului. Conectarea componentei la dispozitivul OBD-II de scanare a autovehiculului are loc folosind comunicații Bluetooth cu ajutorul unei interfețe prin care utilizatorul selectează adresa MAC a dispozitivului OBD-II, respectiv adresa IP a server-ului Euphoria.

¹ Acronim pentru Unified Modeling Language.

3. Aplicație pentru transferul automat a datelor între dispozitive smart personale și sistemul rulând în interiorul autovehiculului

Pentru demonstrarea transferului automat de conținut digital între dispozitivele personale ale șoferului și sistemul rulând în interiorul autovehiculului, am implementat în cadrul etapei o aplicație software pentru un ceas smart (Samsung Gear Fit 2²) și pe un dispozitiv conectat reprezentat de o tabletă instalată în autovehicul, care prezintă o interfață utilizator cu trei componente: (UI₁) informații privind utilizatorul/șoferul cu patru secțiuni (UI_{1.1}) identificarea utilizatorului, (UI_{1.2}) raportarea de parametri fiziologici (bătăile inimii) în timp interactiv, (UI_{1.3}) raportarea de parametri cinematici capturați în interiorul mașinii privind mișcarea mâinilor în timpul condusului, respectiv (UI_{1.4}) parametri cinematici capturați în exteriorul mașinii (numărul de pași) și transferați automat pentru afișare în interfeța utilizator din interiorul vehiculului; (UI₂) informații privind lista de melodii preferate ale utilizatorului care sunt stocate pe ceasul smart; respectiv (UI₃) informații privind lista sarcinilor de efectuat (to-do list), de asemenea stocată pe ceasul smart. Toate informațiile sunt transferate de pe ceas și actualizate atât cât timp ceasul este conectat la sistemul ce afișează interfața: parametrii fiziologici și cinematici sunt actualizați la fiecare secundă (actualizare automată) iar informațiile (UI₂) și (UI₃) cu ocazia evenimentelor utilizator detectate de interfețele celor două dispozitive: fie prin alegerea unei melodii folosind un gest de atingere pe ecranul ce afișează interfața utilizator, fie prin marcarea unei sarcini pe ceas ca fiind îndeplinită.

Dezvoltarea aplicației a fost realizată folosind tehnologii JavaScript, comunicații WebSockets [WebSockets] pe HTTP și node.js [NodeJS].

4. Modelarea conceptuală a mesajelor și asigurarea persistenței informațiilor în interiorul autovehiculului

În prima versiune a implementării noastre, descrisă în raportul de cercetare-dezvoltare aferent Etapei nr. 1 de implementare a proiectului de cercetare-dezvoltare, mesajele în format JSON transmise către arhitectura Euphoria [Schipor *et al.*, 2019] respectau următoarea structură:

- Câmpul *Header* cuprinde informații privind tipul evenimentului și numele dispozitivului sau al aplicației care a produs evenimentul respectiv.
- Câmpul *Conținut* cuprinde informațiile aferente evenimentului produs. Exemple pentru diverse aplicații practice sunt ilustrate în lucrarea Schipor *et al.* (2019).

² <https://www.samsung.com/ro/wearables/gear-fit2-pro/SM-R365NZKNROM/>

Având în vedere faptul că pot exista o varietate de producători de mesaje transmise către Euphoria în contextul auto, precum și de faptul că numărul de producători poate varia în timp, am procedat la identificarea și modelarea conceptuală a diverselor tipuri de producători și a datelor ce pot fi furnizate pentru a determina exact formatul acestor mesaje. Astfel, am identificat și propus în cadrul etapei următorii producători de mesaje Euphoria:

- **Dispozitivul OBD-II (On Board Diagnostics Information)**, citește de la autovehicul și transmite către Euphoria informații privind starea autovehiculului.
- **Aplicația software ANOTIVE (Adaptive NOTifications In-VEhicle)**, rulează pe telefonul smart al șoferului, comunicând cu arhitectura Euphoria.
- **Aplicația TRAFFIC-WEATHER**, rulează pe telefonul smart al șoferului.
- **Ceas smart** pentru achiziția și transmiterea de comenzi vocale.
- **Controller-ul Leap Motion**, un dispozitiv bazat pe tehnici de viziune artificială și stereoviziune folosit pentru achiziția și interpretarea gesturilor mainii, informații ce sunt trimise arhitecturii Euphoria pentru procesare și control.

Pentru a modela datele furnizate de către diverse categorii de producători, am aplicat un mecanism de abstractizare ce ne-a permis centrarea pe datele de interes în vederea funcționării corecte a diverselor aplicații dezvoltate. De exemplu, din totalitatea informațiilor achiziționate și transmise de către dispozitivul OBD-II, suntem interesați în principal de numele funcției, descrierea, modul, PID-ul, valoarea minimă și maximă, respectiv de unitatea de măsură corespunzătoare.

Datele existente în sistemul in-vehicle pot fi *structurate* și *semi-structurate*. Am propus în cadrul etapei un mecanism de persistență a acestor date folosind un model NoSQL MongoDB [MongoDB].³ Controlul accesului la date este realizat la nivel de acces la baza de date.

Diversitatea formatelor de date și dimensiunea datelor prevăzute a fi stocate sunt de natură să necesite proiectare specifică a schemei de date [Imam *et al.*, 2018; William, 2014]. Astfel, deși în general bazele de date NoSQL sunt utilizate într-o manieră *schema-free*, există diverse opțiuni de proiectare a depozitelor de date, astfel încât stocarea să fie optimizată în raport cu modul în care vor fi interogate și utilizate datele ulterior [MongoDB; Imam *et al.*, 2018]. Informațiile vehiculate în arhitectura Euphoria între producători și consumatori sunt stocate în

³ MongoDB aparține sistemelor de gestiune a bazelor de date bazate pe documente. Informațiile stocate reprezintă colecții eterogene ierarhice denumite *documente*. Acestea sunt formate din perechi de tip cheie-valoare, fiecare cheie specificând un câmp din cadrul documentului. Perechile pot fi incluse în tabele asociative, serializate ca și documente JSON, respectiv stocate pe disc în format BSON (Binary JavaScript Object Notation).

format brut ca și colecții indexate temporal. Ulterior, aceste date vor putea fi procesate prin interogări specifice MongoDB. Arhitectura datelor este flexibilă, scalabilă și oferă operațiuni de sortare, indexare și interogări similare bazelor relaționale [Franciscus *et al.*, 2018].

Interacțiunea dintre sistemul in-vehicle și nivelul de persistență a datelor se realizează prin intermediul a două componente software, implementate în node.js [NodeJS]:

- **MongoLogger** permite traficul datelor brute între producători, prin intermediul Euphoria, către depozitul de date MongoDB.
- **MongoReader** permite traficul datelor brute dinspre depozitul de date către consumatorii care au emis cereri pentru astfel de date.

Din punct de vedere comportamental, componentele MongoLogger și MongoReader sunt descrise în diagramele de secvențe din figurile 9 și 10. Datele brute, recepționate de consumatori, sunt prelucrate la nivelul acestora. Diagramele UML au fost realizate folosind IDE-ul Astah [Astah].

6. Referințe

[Astah] Astah IDE: <http://astah.net/>

[Franciscus *et al.*, 2018] Franciscus, N., Ren, X., & Stantic, B. (2018). Precomputing architecture for flexible and efficient big data analytics. *Vietnam Journal of Computer Science*, 5(2), 133-142. <https://link.springer.com/article/10.1007/s40595-018-0109-9>

[Imam *et al.*, 2018] Imam, A. A., Basri, S., Ahmad, R., Watada, J., & González-Aparicio, M. T. (2018). Automatic schema suggestion model for NoSQL document-stores databases. *Journal of Big Data*, 5(1), 46. , <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-018-0156-1#Sec3>

[Lourenço *et al.*, 2015] Lourenço, J. R., Cabral, B., Carreiro, P., Vieira, M., & Bernardino, J. (2015). Choosing the right NoSQL database for the job: a quality attribute evaluation. *Journal of Big Data*, 2(1), 18. <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-015-0025-0>

[MongoDB] MongoDB, <https://www.mongodb.com/>.

[NodeJS] Node.js, <https://nodejs.org/en/>

[Schipor *et al.*, 2019] Ovidiu-Andrei Schipor, Radu-Daniel Vatavu, Jean Vanderdonckt. (2019). Euphoria: A Scalable, Event-Driven Architecture for Designing Interactions Across heterogeneous Devices in Smart Environments. *Information and Software Technology* 109. Elsevier, 43-59. <https://doi.org/10.1016/j.infsof.2019.01.006>

[UML] OMG: Unified Modelling Language Superstructure, version 2.0, ptc/03-0802, 2003.

[WebSockets] WebSockets API, https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API

[William, 2014] William Z. 6 Rules of thumb for MongoDB schema design. MongoDB, 2014. <https://www.mongodb.com/blog/post/6-rules-of-thumb-for-mongodb-schema-design-part-1> Accessed: 11.11.2019