UNIVERSITATEA „BABEȘ - BOLYAI” CLUJ-NAPOCA
ȘCOALA DOCTORALĂ DE ȘTIINȚE ECONOMICE ȘI GESTIUNEA AFACERILOR

TEZĂ DE DOCTORAT

-Rezumat-

Procesarea semantică la marginea rețelei a datelor senzor din mediile inteligente: o abordare bazată pe paradigma Design Science

Conducător de doctorat:
Prof. univ. dr. Tomai Nicolae

Student-doctorand:
Zălhan Paula-Georgiana

Cluj-Napoca

2020

Cuprinsul rezumatului

1. Introducere 1

2. Procesarea fluxului de date în IoT 5

3. Tehnologii semantice pentru domeniul IoT 8

4. Stadiul cunoașterii în Procesarea Semantică a Fluxului de Date 11

5. Proiectarea și implementarea sistemului de tip ”pipeline” pentru procesarea semantică a fluxului de date 13

6. Cazuri de utilizare IoT în mediile inteligente 19

7. Concluzii și direcții viitoare de cercetare 23

Cuvinte cheie

Internetul lucrurilor (IoT), Edge computing, Procesarea semantică a fluxului de date, date senzor, adnotare semantică, ontologia Semantic Sensor Network (SSN), Apache Kafka, metodologia de cercetare Design Science.

Cuprinsul tezei de doctorat

1. Introducere

1.1. Definirea problemei

1.2. Întrebări de cercetare

1.3. Contribuții la această cercetare

1.4. Structura lucrării

2. Procesarea fluxului de date în IoT

2.1. Fundament teoretic cu privire la procesarea fluxului de date în IoT

2.1.1. Termenul „flux de date”

2.1.2. Paradigma de procesare a fluxului de date

2.1.3. Caracteristicile fluxului de date în IoT

2.2. Arhitecturi bazate pe evenimente pentru analiza în timp real a Big Data

2.2.1. Arhitectura Lambda

2.2.2. Arhitectura Kappa

2.3. Protocoale și standarde în IoT

2.4. Arhitectura generală a unui sistem IoT de tip „pipeline” pentru procesarea Big Data

3. Tehnologii semantice pentru IoT

3.1. Modelul „Resource Description Framework”

3.2. Terminologia standard „RDF Schema”

3.3. Terminologia standard „Web Ontology Language”

3.3.1. Clase

3.3.2. Proprietăți

3.3.3. Afirmații despre lucruri

3.4. Ontologia „Semantic Sensor Network”

4. Stadiul cunoașterii în Procesarea Semantică a Fluxului de Date

4.1. Procesarea fluxului de date RDF

4.1.1. Sisteme centralizate de procesare a fluxului de date RDF

4.1.2. Sisteme distribuite de procesare a fluxului de date RDF

4.1.3. Soluții middleware

4.2. Raționare pe fluxuri de date

5. Proiectarea și implementarea sistemului de tip „pipeline” pentru procesarea semantică a fluxului de date

5.1. Metodologia adoptată

5.2. Prezentarea generală a soluției propuse: un sistem de tip „pipeline” pentru procesarea semantică a fluxului de date

5.2.1. Surse de date

5.2.2. Colectarea și procesarea fluxului de date

5.2.3. Stocarea datelor

5.2.4. Analiza datelor

5.3. Extensii propuse pentru ontologia SSN

5.4. Implementarea soluției

5.4.1. Configurarea clusterului Kafka

5.4.1.1. Configurarea un nod – un broker

5.4.1.2. Configurarea un nod – mai mulți brokeri

5.4.2. Arhitecturile alternative ale sistemului de tip „pipeline”

5.4.2.1. Scrierea producătorilor

5.4.2.2. Scrierea consumatorilor

5.5. Validarea soluției

6. Cazuri de utilizare IoT în mediile inteligente

6.1. Introducere

6.2. Aeroport inteligent

6.2.1. Descrierea scenariului

6.2.2. Decizii de proiectare

6.2.3. Scenarii experimentale și rezultate obținute

6.2.3.1. Scenarii experimentale

6.2.3.2. Analiza rezultatelor

6.3. Fabrică inteligentă

6.3.1. Descrierea scenariului

6.3.2. Decizii de proiectare

6.3.3. Scenarii experimentale și rezultate obținute

6.3.3.1. Scenarii experimentale

6.3.3.2. Analiza rezultatelor

7. Concluzii și direcții viitoare de cercetare

Anexa A. Cod sursă pentru Producător de fluxuri de date RDF

Anexa B. Cod sursă pentru Consumator de fluxuri de date RDF

Referințe bibliografice

Lista publicațiilor proprii

I. Lista publicațiilor referitoare la cuprinsul tezei de doctorat

1. **P.-G. Zălhan**, G. C. Silaghi and R. A. Buchmann. *Marrying Big Data with Smart Data in Sensor Stream Processing*. În A. Siarheyeva, C. Barry, M. Lang, H. Linger, & C. Schneider (Eds.), Information Systems Development: Information Systems Beyond 2020 (ISD2019 Proceedings). ISD'19. Toulon, France: ISEN Yncréa Méditerranée: AIS eLibrary. URL: https://aisel.aisnet.org/isd2014/proceedings2019/ManagingISD/8/.
2. **P. -G. Zălhan**, *Transforming Big Data into knowledge using semantic stream processing technology: challenges and early progress*. În Proceedings of the 18th International Conference on INFORMATICS in ECONOMY (IE 2019) Education Research & Business Technologies, 2019, pp. 365-370, ISSN 2284-7472.

II. Lista altor publicații relevante din stagiul doctoral

1. **P.-G. Zălhan**, *Building a LVCSR System for Romanian: methods and challenges*, Journal of Public Administration, Finance and Law (JoPAFL), nr. 10/2016, pp. 181-191, 2016, ISSN 2285 – 2204.
2. C.-C. Osman, **P.-G. Zălhan**, *From Natural Language Text to visual models: A survey issues and approaches*, Revista Informatică Economică, vol. 20, nr. 4, pp. 44-61, 2016, ISSN 1453-1305.

1. Introducere

În era digitală din ziua de astăzi, se remarcă o creștere exponențială a datelor transmise de Internetul lucrurilor (eng. Internet of Things - IoT), date care sunt generate cu repeziciune în mediile inteligente. Această creștere exponențială a datelor IoT este cauzată de adoptarea la scară mare a paradigmei IoT, în care numărul dispozitivelor inteligente conectate la Internet continuă să crească într-un ritm constant. Statisticile referitoare la datele IoT, statistici ce sunt furnizate de marile organizații de cercetare precum Cisco, arată că *5 cvintilioane de octeți de date sunt produse în fiecare zi* (Stack, 2018). Mai mult, creșterea dramatică a datelor IoT este anticipată de organizația Cisco, care preconizează că *până la sfârșitul anului 2030 vor fi 500 de miliarde de dispozitive IoT conectate la Internet* (Cisco, 2018).

Chiar dacă domeniul IoT este considerat unul dintre cele mai interesante domenii de cercetare datorită caracterului său inovativ pe care îl deține, totuși complexitatea acestui domeniu de cercetare ridică niște provocări semnificative care ar putea sta în calea realizării beneficiilor sale potențiale. Un studiu recent (Noura, Atiquzzaman, & Gaedke, 2019) legat de interoperabilitatea aplicațiilor IoT, arată că problema interoperabilității nu este încă depășită, chiar dacă de-a lungul timpului oamenii de știință au propus diferite metode și tehnologii menite să diminueze această problemă. De asemenea, interoperabilitatea la nivel semantic este cea mai puțin tratată de oamenii de știință în soluțiile propuse de aceștia. Tot în acest studiu, autorii afirmă că soluțiile IoT existente nu consideră o soluție de *edge computing* pentru a mări viteza de transfer a datelor și pentru a eficientiza analiza acestora.

Seturile masive de date, numite generic „Big Data”, trebuie procesate, stocate și prezentate într-o formă eficientă într-un interval de timp scurt. Dacă la început, ecosistemul Big Data presupunea o colecție de metode menite să gestioneze volumul de date, în prezent, trebuie să se ia în considerare și problema vitezei, prin dezvoltarea unor sisteme capabile să colecteze și să proceseze fluxul de date în timp real. Astfel de sisteme sunt necesare pentru a îmbunătăți procesele de luare a deciziilor, însă în proiectarea unor astfel de sisteme trebuie să se țină cont de particularitățile fluxurilor de date care îngreunează analiza lor.

Natură eterogenă a fluxului de date provenit din varii surse conduce la probleme de interoperabilitate între aplicațiile IoT iar pentru a permite experților în domeniu sau a oamenilor de afaceri să ia decizii corecte, în mod eficient, este necesară dezvoltarea unor tehnologii care să extragă informații utile din datele generate din mediile inteligente. În acest sens, comunitatea de Web Semantic și-a îndreptat eforturile de cercetare în construirea unor modele și tehnici de procesare menite să transforme datele capturate de la dispozitivele inteligente (ex. senzori) în reprezentări inteligibile, ce permit descrierea obiectului care a generat acele date, partajarea și integrarea informației, precum și generarea de noi cunoștințe despre mediile fizice în care au fost capturare fluxurile de date. Prin *adnotare semantică*, care este procesul de atașare a informațiilor suplimentare la diverse concepte (ex. oameni, lucruri, locuri, evenimente, numere, etc.) dintr-un text sau un conținut – datele colectate de la senzori pot fi transformate în descrieri prelucrabile care sunt ușor de interpretat, integrat și reutilizat de către mașini. Conceptele identificate și îmbogățite semantic pot fi apoi stocate într-o bază de date semantică (numit și *depozit de triplete*) pentru executarea unor inferențe asupra datelor cu rolul de a deduce noi cunoștințe.

Scopul acestui studiu de cercetare cantitativă este de a construi un sistem care e capabil să colecteze, să proceseze, să stocheze și să prezinte volumele mari de date IoT într-o manieră eficientă și ușor de interpretat. Un sistem de tip „pipeline” acoperă toate aspectele unui lanț integral de prelucrare a datelor, din momentul în care datele au fost generate, până în momentul când datele sunt pregătite pentru consumul final. Mutarea datelor de la surse la un depozit de date, precum și analiza acestora trebuie realizate în mod consistent și eficient. În retrospectivă, această cercetare își propune să proiecteze un sistem de tip „pipeline”, scalabil, capabil să gestioneze un debit mare de date în timp real.

Pentru a atinge scopul prezentei lucrări de cercetare, paradigma de cercetare Design Science (eng. Design Science Research - DSR) a fost adoptată deoarece aceasta este considerată a fi potrivită atunci când cercetarea este axată pe soluționări de probleme detectate în contextul de cercetare. Soluțiile propuse se numesc artefacte și rezolvă probleme specifice unor Sisteme Informaționale (SI), propunând noi soluții sau îmbunătățind soluții existente. Problema de proiectare pe care această cercetare încearcă să o depășească este problema interoperabilității semantice în domeniul IoT prin dezvoltarea unei soluții inovative de tip „pipeline” care să îndeplinească cerința de latență scăzută în procesul de analiză a datelor pentru a extrage informații

oportune și valoroase din date, informații ce sunt necesare pentru îmbunătățirea procesului decizional.

În vederea îndeplinirii scopului prezentei teze, această cercetare adresează următoarele întrebări de cercetare:

1. Cum poate fi conceput un sistem de tip „pipeline” cu latență scăzută pentru procesarea semantică a fluxului de date, sistem care să fie capabil de a susține îmbogățirea semantică a volumelor mari de date de la senzorii instalați într-un mediu inteligent?
2. Putem să identificăm locul preferabil pentru executarea sarcinii de adnotare semantică a fluxului de date într-un sistem de tip „pipeline” în vederea unei analize mai rapide a datelor?
3. Cum poate fi proiectat și administrat un mediu de tip „semantic edge” pentru a susține procesarea în timp real a datelor și execuția fluxului de lucru a datelor senzor?

Pentru a răspunde la aceste întrebări de cercetare, următoarele obiective trebuie îndeplinite:

1. Identificarea cerințelor și a limitărilor din literatura științifică actuală din domeniul procesării semantice a fluxului de date.
2. Obținerea interoperabilității semantice cu SI care se bazează pe fluxurile de date adnotate.
3. Evaluarea performanței sistemului propus de tip „pipeline” prin cazuri reale de utilizare din domeniul IoT.
4. Propunerea unor direcții viitoare de cercetare și îmbunătățiri care sunt semnificative pentru a continua dezvoltarea sistemului de tip „pipeline”.

Prin urmare, teza de față aduce următoarele contribuții în ceea ce privește tematica de procesare semantică a fluxului de date:

1. Furnizează o revizuire a literaturii de specialitate cu privire la procesarea semantică a fluxurilor de date reliefând eforturile științifice care au fost depuse la intersecția procesării semantice în IoT și a tehnologiilor semantice existente în acest sens.
2. Extinde ontologia Semantic Sensor Network (SSN) identificată în literatură cu noi termeni, dintre care unii sunt independenți de scenariu, iar alții țintesc cazuri specifice din domeniul IoT. Astfel, clasele propuse creează o ierarhie de tipuri specifice de senzori, de observații, de obiecte-de-interes cu caracteristici măsurabile, iar proprietățile

introduse descriu relații între senzori și observațiile lor critice, relații între senzori și entitățile care îi găzduiesc, precum și relații între diferite entități din mediul inteligent.

3. Proiectează o arhitectură pentru artefactul propus, arhitectură care deține mai multe componente: colectarea datelor de la senzori, adnotarea semantică a fluxurilor de date senzor, stocarea fluxurilor de date îmbogățite semantic într-un depozit de triplete, precum și executarea unor interogări asupra acestor date în vederea integrării acestora și a deducerii de noi cunoștințe.
4. Realizează o dovadă de concept (eng. Proof of Concept) pentru arhitectura sistemului de tip „pipeline" demonstrând faptul că procesarea semantică la marginea rețelei poate fi realizată în faza de dezvoltare a artefactului prin procesul iterativ al DSR.
5. Validează prototipurile rezultate în urma fiecărei iterații din procesul DSR utilizând două scenarii diferite de afaceri.

Continuarea prezentei teze este organizată după cum urmează:

Capitolul 2 descrie conceptele legate de procesarea semantică a fluxului de date din domeniul IoT. Acest capitol explică noțiunea de „flux de date" și caracteristicile sale, apoi descrie paradigma de procesare a fluxului de date, precum și arhitecturile existente de procesare a fluxului de date. Pe lângă lista de protocoale specifice aplicațiilor IoT, acest capitol prezintă arhitectura generală a unui sistem de tip „pipeline" pentru procesarea Big Data.

Capitolul 3 prezintă aspecte teoretice cu privire la standardele și modelele existente de Web Semantic pentru adnotare semantică, punându-se accent pe termenii furnizați de ontologia SSN.

Capitolul 4 furnizează o revizuire a literaturii științifice evidențiind abordările și soluțiile propuse de cercetători în vederea construirii unui sistem de procesare semantică a fluxului de date.

Capitolul 5 descrie metodologia adoptată, sistemul de tip „pipeline" propus, metricile de performanță utilizate pentru a valida acest sistem, precum și extensiile propuse ale ontologiei SSN.

Capitolul 6 prezintă deciziile de proiectare a unui aeroport inteligent și a unei fabrici inteligente și discută rezultatele obținute în evaluarea performanței sistemului de tip „pipeline".

În cele din urmă, în capitolul 7, prezentăm câteva concluzii și sunt descrise direcțiile de cercetare viitoare.

2. Procesarea fluxului de date în IoT

Acest capitol prezintă fundamentele teoretice cu privire la procesarea fluxurilor de date. Conceptele definite în cele ce urmează vor fi utilizate în capitolul 5 pentru a proiecta sistemul de tip „pipeline” pentru procesarea semantică a fluxului de date. Este deosebit de important de a alege arhitectura potrivită de procesare a datelor precum și a ști care sunt etapele principale în construirea unui sistem de tip „pipeline”.

2.1. Fundament teoretic al procesării fluxului de date în IoT

Paradigma de procesare a fluxului de date (Liu, Dastjerdi, & Buyya, 2016) a fost propusă de comunitățile de cercetare ca soluție pentru nevoia de a procesa fluxuri masive de date, într-un timp limitat, utilizând resurse distribuite. Spre deosebire de paradigma de procesare batch a datelor care e incapabilă de a gestiona data dinamice generate cu o viteză ridicată în mediile inteligente, paradigma de procesare a fluxului de date este soluția ideală pentru a gestiona ecosisteme IoT întrucât are abilitatea de a analiza în timp real datele provenite din surse diferite.

Totuși, procesarea în timp real a acestor date este una dificilă întrucât trebuie să se țină cont de caracteristicile fluxurilor de date, precum dinamism, imperfecțiuni, continuitate, eterogenitate, volatilitate. Un sistem de procesare a fluxurilor de date trebuie sa fie capabil de a procesa datele în mișcare, de îndată ce aceste devin disponibile pentru a răspunde la dinamismul mediului IoT. Deoarece fluxurile de date sunt procesate în timp util, infrastructura trebuie să facă față imperfecțiunilor fluxului de date: date întârziate, date lipsă sau date care nu sosesc în aceeași ordine (Stonebraker, Cetintemel, & Zdonik, 2005). Fluxul de date, care e o secvență infinită de elemente de date care variază în timp, traversează în mod continuu prin sistemul de procesare de tip „pipeline” și este actualizat de fiecare dată când devine disponibil. Mai mult, fluxul de date poate proveni din surse diferite și poate exista în diferite formate (audio, video, text, etc.).

2.2. Arhitecturi bazate pe evenimente pentru analiza în timp real a Big Data

Spre deosebire de arhitecturile bazate pe servicii care restricționează scalabilitatea unui sistem, datorită mecanismului sincronizat de cerere-răspuns între producător și consumator (Jerry, 2019),

arhitecturile bazate pe evenimente gestionează în mod asincron fluxurile de date, fiind deseori utilizate în construirea sistemelor scalabile și adaptabile la schimbările mediului. Întrucât un flux de date este echivalent cu un eveniment care are loc într-un mediu inteligent, există două tipuri de arhitecturi bazate pe evenimente care sunt capabile de a gestiona volume mari de date: arhitectura Lambda și arhitectura Kappa. Dacă arhitectura Lambda (Marz & Warren, 2015) folosește în combinație analiza batch cu analiza în timp real a evenimentelor, arhitectura Kappa (Kreps, 2014) vine ca o simplificare a celei dintâi, eliminând nivelul de batch. De obicei, arhitectura Kappa este utilizată pentru a proiecta și implementa aplicații care trebuie să analizeze în timp real evenimentele complexe generate în mediile inteligente.

2.3. Protocoale și standarde în IoT

Protocoalele standard specifice mediilor IoT asigură comunicarea datelor între dispozitivele conectate la Internet, iar printre aceste protocoale se numără:

- *Message Queuing Telemetry Transport* (MQTT) – protocol pentru comunicarea mașină către mașină în mediile edge a dispozitivelor cu putere redusă.
- *Message Queuing Telemetry Transport-Sensor Network* (MQTT-SN) – versiune a MQTT special destinată pentru rețelele de senzori fără fir, unde dispozitivele au baterie limitată (Govindan & Azad, 2015; Stanford-Clark & Truong, 2013).
- *Constrained Application Protocol* (CoAP) – protocol pentru comunicarea între dispozitivele dintr-o rețea constrânsă, cu lățime de bandă redusă și disponibilitate scăzută.
- *Advanced Message Queuing Protocol* (AMQP) – protocol pentru soluții middleware (Firouzi, Chakrabarty, & Nassif, 2020) orientate pe schimb de mesaje.

2.4. Arhitectura generală a unui sistem IoT de tip „pipeline” pentru procesarea Big Data

Un sistem de tip „pipeline” pentru procesarea volumelor mari de date trebuie să fie capabil de a captura, transforma, stoca și prezenta datele relevante, în timp real. În Figura 1 este prezentată arhitectura generală a unui sistem de tip „pipeline” pentru procesarea Big Data, sistem care conține mai multe componente: ingerarea datelor, procesarea datelor, stocarea datelor, precum și vizualizarea datelor.

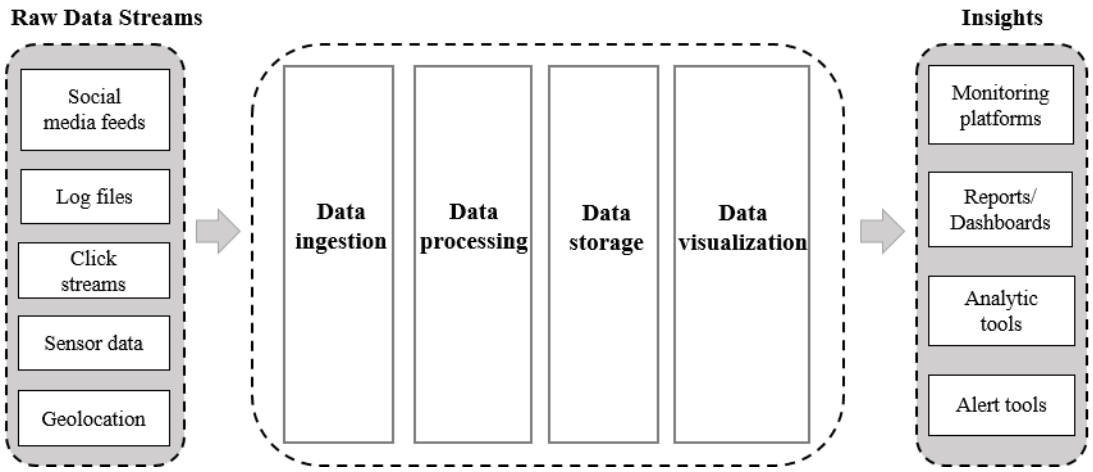

Figura 1. Arhitectura generală a unui sistem de tip „pipeline" pentru procesarea Big Data *(Zălhan, 2019)*

În etapa de ingerare a datelor, fluxurile de date sunt colectate din surse diferite, precum rețele de socializare, rețele de senzori fără fir sau alte dispozitive IoT. Apoi, pe măsură ce aceste date intră în sistem, ele suferă transformări complexe cu ajutorul unor operatori specifici de filtrare, agregare și unire. Aceste procesări pot fi restricționate ca să aibă loc doar asupra unei porțiuni finite din fluxul de date, porțiune numită fereastră. Pentru a susține o analiză în timp real a datelor, cu un timp de răspuns cât mai mic, fluxurile de date pot fi procesate în paralel cu ajutorul mai multor mașini de calcul din cadrul unui cluster. Etapa de stocare are rolul de a persista datele relevante într-o bază de date pentru o analiză ulterioară, mai complexă, menită să extragă informații utile din fluxul de date. Etapa de vizualizare sau de prezentare are rolul de a crea o interfață a sistemului pentru utilizatorii umani. Această interfață poate fi un panou de control pentru monitorizarea valorilor în timp real a datelor, sau un instrument de alertă pentru notificarea utilizatorilor în cazul detectării unor valori critice în mediul inteligent.

3. Tehnologii semantice pentru domeniul IoT

Acest capitol prezintă fundamental teoretic cu privire la standardele și modelele de Web Semantic. Conceptele definite în cele ce urmează vor fi utilizate în capitolele 5 și 6 pentru procesul de adnotare semantică a fluxurilor de date senzor care traversează sistemul de tip „pipeline". Îmbogățirea semantică a datelor generate de dispozitivele inteligente cu informații adiționale din surse externe, precum vocabulare sau ontologii, contribuie la interoperabilitatea aplicațiilor IoT. Furnizarea unor descrieri semantice a datelor senzor ajută ca mașinile de calcul să înțeleagă într-un mod unitar contextul în care aceste date au fost generate.

3.1. Modelul „Resource Description Framework"

Modelul Resource Description Framework (RDF) (Klyne, Carroll, & McBride, 2009) ajută la crearea unor reprezentări bazate pe grafuri a unor resurse din Web, folosind o varietate de notații sintactice și formate de serializare a datelor. Structura internă a modelului de date RDF constă în expresii de forma: subiect-predicat-obiect, numite *triplete* (Domingue, Fensel, & Hendler, 2011), unde un predicat descrie relația dintre un subiect și un obiect. O colecție de triplete formează un graf orientat, unde nodurile RDF sunt subiectele și obiectele sale. Pentru a stoca grafurile RDF, există mai multe formate de serializare, cum ar fi, formatul Turtle. Vocabularul RDF conține un set minimal de termeni care pot fi utilizați pentru a construi afirmații RDF despre relațiile dintre diferite resurse.

3.2. Terminologia standard „RDF Schema"

Vocabularul RDF Schema (RDFS) extinde vocabularul standard RDF cu clase și proprietăți adiționale ce permit descrierea înțelesului unor resurse, precum și relațiile dintre acestea. Aceste resurse sunt folosite pentru a determina caracteristicile altor resurse, cum ar fi: domeniul și codomeniul unor proprietăți. Predicatele standard ″rdfs:domain″ (domeniu) și ″rdfs:range″ (codomeniu) pot fi utilizate pentru a specifica restricții între subiectele și obiectele din tripletele RDF. Proprietățile standard ″rdfs:subClassOf″ și ″rdfs:subPropertyOf″ pot fi utilizate pentru a construi ierarhii de clase, respectiv ierarhii de proprietăți. Există și alte vocabulare RDF care sunt

deseori folosite pentru a construi aplicații de Web Semantic. Spre exemplu, vocabularul Schema.org deține un set de termeni predefiniți pentru a descrie concepte precum Persoană, Angajat, Organizație, etc.

3.3. Terminologia standard „Web Ontology Language”

Vocabularul RDFS furnizează o semantică simplă pentru aplicațiile IoT, permițând definirea unor clase ce dețin la rândul lor alte subclase, și proprietăți care au multiple subproprietăți, domenii și codomenii. Terminologia standard Web Ontology Language (OWL) (Motik, și alții, 2009) extinde RDFS cu o semantică bogată, necesară pentru obținerea interoperabilității între diferite componente ale unui sistem IoT (Li & Zhong, 2004). Termenii standardizați de tip OWL și RDFS pot fi folosiți pentru a construi *axiome*, adică noi afirmații RDF care au subiecte clase sau proprietăți. Spre exemplu, proprietatea standard ″owl:inverseOf″ furnizată de OWL poate fi folosită pentru a defini relații inverse între proprietăți, adică definirea unor relații în ambele direcții.

3.4. Ontologia „Semantic Sensor Network”

Ontologia Semantic Sensor Network (SSN) (Haller, și alții, 2017) este o tehnologie semantică cheie care facilitează interoperabilitatea semantică, integrarea și raționarea datelor senzor. Această ontologie se focalizează pe descrierea rețelelor fizice de senzori, furnizând informații despre senzori, observații care rezultă în urma detectării unor noi valori senzor, platformele care găzduiesc senzori, etc. Ontologia SSN include o ontologie de bază, de sine stătătoare, numită SOSA (Sensor, Observation, Sample, and Actuator) (Janowicz, Haller, Cox, Le Phuoc, & Lefrançois, 2019), care furnizează clasele și proprietățile de bază ale ontologiei SSN. Pentru modelarea observațiilor senzorilor instalați în mediile inteligente, dar și a altor concepte asociate, următorii termeni SOSA sunt utilizați:

- *sosa:Sensor* – clasă ce reprezintă conceptul de Senzor.
- *sosa:Platform* – conceptul de Platformă este entitatea care găzduiește alte entități, precum senzori.
- *sosa:FeatureOfInterest* – conceptul de Obiect-de-Interes a cărui proprietate este calculată când rezultatul unei observații a fost transmis în mediul inteligent de către senzor.
- *sosa:ObservableProperty* – conceptul de Caracteristică Observabilă a obiectului-de-interes.

- *sosa:Observation* – conceptul de Observație reprezintă activitatea de a estima sau de a calcula o valoare a proprietății obiectului-de-interes.
- *sosa:Result* – conceptul de Rezultat reprezintă rezultatul unei observații generate în mediu.
- *sosa:Procedure* – noțiunea de Procedură care descrie un flux de lucru, un algoritm sau o metodă computațională care specifică cum este generată o observație în mediul inteligent, sau care sunt pașii care trebuie executați pentru a ajunge la un rezultat.

Proprietățile SOSA sunt definite în ambele direcții, dinspre domeniu, înspre codomeniu. Spre exemplu, proprietatea standard ″sosa:madeObservation″ descrie relația dintre un senzor și observația făcută de acesta, pe când proprietatea inversă ″sosa:madeBySensor″ conectează observațiile cu senzorii ce au generat acele observații. Pentru a modela valoarea observației generate în mediul IoT, proprietatea standard ″sosa:hasResult″ poate fi utilizată. Această proprietate leagă o observație de rezultatul care conține valoarea observată. Proprietatea ″sosa:hasResult″ este folosită pentru a modela informații complexe despre o observație, informații precum: ce măsoară un senzor în mediu, care este unitatea de măsură a valorii senzorului, sau unde este localizat senzorul în mediul inteligent.

Ontologia SSN, fiind construită deasupra ontologiei SOSA, introduce termeni adiționali, precum: ″ssn:Stimulus″, ″ssn:Input″, ″ssn:Output″, ″ssn:System″, ″ssn:Deployment″, pentru modelarea stimulilor exteriori, a datelor de intrare sau de ieșire a unei proceduri, a echipamentelor de infrastructură care implementează proceduri, a modalităților de implementare a unor astfel de echipamente.

Conform cu raportul final al World Wide Web Consortium (W3C), autorii (Vadivel & Subramanian, 2017) afirmă că ontologia SSN are câteva limitări:

- Ontologia SSN exclude modelarea unei ierarhii de senzori specifici instalați în mediile inteligente, precum și ierarhii de tipuri specifice de obiecte-de-interes.
- Termeni pentru reprezentarea unităților de măsură asociate valorilor senzorilor, dar și termeni pentru locația senzorilor în mediu, de asemenea, nu sunt incluși în ontologie.

4. Stadiul cunoașterii în Procesarea Semantică a Fluxului de Date

Pe baza unei revizuiri a literaturii de specialitate cu privire la procesarea semantică a fluxurilor de date, în acest capitol dezvoltăm un cadru teoretic pentru a înțelege și analiza eforturile depuse în domeniul de Procesare Semantică a Fluxurilor de Date și pentru a găsi direcții viitoare de cercetare în vederea depășirii limitelor abordărilor existente. Prezenta revizuire sistematică include lucrări din conferințe și jurnale științifice de mare impact.

Procesarea Semantică a Fluxului de Date este definită de (Le Phuoc & Hauswirth, 2019) ca fiind un set de modele, principii și tehnici folosite pentru analizarea și procesarea fluxurilor de date în vederea explorării structurii și înțelesului elementelor de date din fluxul de date. Focalizându-ne pe progresul științific realizat în domeniul de procesare semantică a fluxului de date, cele mai consolidate rezultate ale cercetării includ: (1) extensii ale modelului RDF și ale limbajului de interogare SPARQL, extensii care sunt produse și implementate de sisteme de procesare a fluxului de date RDF, și (2) tehnici de inferență propuse sub eticheta de „Raționare pe fluxului de date" pentru generarea de noi cunoștințe din fluxurile de date RDF.

În cele ce urmează vom evidenția limitările identificate în tendințele de cercetare menționate mai sus. Contribuția științifică principală a prezentei teze va fi aceea de a depăși aceste limitări, propunând o soluție alternativă.

4.1. Procesarea fluxului de date RDF

Subdomeniul de cercetare „Procesarea fluxului de date RDF", numit și „Procesarea fluxului de Linked Data" (Le-Phuoc, Parreira, & Hauswirth, 2012) propune sisteme de procesare semantică care folosesc modelul RDF în reprezentarea fluxurilor de date și execută interogări SPARQL continue asupra fluxurilor de date RDF, utilizând diferite tehnici.

Primele sisteme de procesare a fluxului de date RDF au fost sistemele centralizate. Aceste sisteme au fost proiectate pentru a rula pe o singură mașină prin aplicarea unor interogări continue asupra fluxurilor de date RDF utilizând extensii SPARQL. Aceste sisteme centralizate extind limbajul SPARQL prin adăugarea unor operatori specifici, precum operatori de agregare, de ferestruire, oferind totodată posibilitatea de combinare a fluxurilor de date. Aceste sisteme sunt

construite pe baza unor Sisteme de Gestiune a Fluxurilor de Date (SGFD) pentru a procesa partea dinamică a interogării, și un depozit de triplete pentru a evalua partea statică a interogării. Totuși, aceste sisteme nu sunt capabile de a gestiona volume mari de date provenite din surse eterogene și nu sunt capabile de a executa multiple interogări continue asupra fluxurilor de date RDF întrucât nu dispun de resurse distribuite pentru a paraleliza analiza datelor.

Sistemele distribuite de procesare a fluxului de date RDF au fost propuse de comunitatea de cercetare pentru a depăși limitările sistemelor centralizate de procesare semantică. Aceste sisteme distribuite de procesare a fluxului de date RDF îmbunătățesc performanța sistemelor existente, folosind o infrastructură de tip cluster pentru procesarea paralelă a fluxurilor de date RDF. Totuși, unele sisteme distribuite se focalizează doar pe date dinamice.

Între timp, au fost propuse mai multe sisteme pentru colectarea în timp real a fluxurilor de date senzor, procesarea și transportarea acestora către sisteme destinație cu ajutorul unei componente de tip middleware. Unele sisteme susțin adnotarea semantică a fluxului de date senzor utilizând termenii standard furnizați de ontologia SSN, iar altele, îmbogățesc semantic aceste fluxuri de date propunând o extensie a ontologiei SSN, extensie ce conține termeni pentru modelarea unităților de măsură asociate unor senzori specifici (temperatură, umiditate, lumină și tensiune). Unele soluții middleware folosesc o infrastructură de tip cloud pentru adnotarea și executarea interogărilor asupra fluxului de date. Totuși, niciun sistem nu propune o soluție de tip edge computing pentru o analiză mai rapidă și mai eficientă a fluxului de date RDF.

4.2. Raționare pe fluxuri de date

Sistemele de raționare pe fluxuri de date extind sistemele de procesare a fluxului de date RDF cu capacități logice, bazate pe reguli. Unele sisteme folosesc limbaje de raționare care extind logicile de ordinul întâi și al doilea cu operatori specifici de ferestruire. Există abordări bazate pe modele de raționare pe fluxurile de date adnotate semantic care susțin luarea deciziilor în mediile inteligente (de exemplu, orașe inteligente). Raționarea pe fluxuri de date poate fi combinată cu tehnici de învățare automată (eng. machine learning), cum ar fi algoritmi de învățare supravegheată pentru clasificarea fluxurilor de date.

5. Proiectarea și implementarea sistemului de tip ”pipeline” pentru procesarea semantică a fluxului de date

În acest capitol descriem metodologia adoptată în construirea sistemului de tip „pipeline” pentru procesarea semantică a fluxului de date. Prezentăm arhitectura sistemului propus evidențiind componentele sale și tehnologiile folosite în proiectarea sa. Descriem termenii noi introduși ca extensii ale ontologiei SSN pentru a modela concepte independente de scenariu, precum și concepte legate de scenarii IoT specifice. Descriem succint configurările realizate în implementarea sistemului, precum și arhitecturile alternative ale sistemului de tip „pipeline” pe care le considerăm atunci când investigăm unde este preferabil de a executa sarcina de adnotare semantică a fluxului de date. În cele din urmă discutăm metricile de performanță luate în considerare pentru a valida soluția propusă.

5.1. Metodologia adoptată

În construirea sistemului de tip „pipeline” pentru procesarea semantică a fluxului de date, este adoptată metodologia de cercetare *Design Science* (Wieringa, 2014), cu ajutorul căreia urmăm un ciclu de dezvoltare iterativă a sistemului propus cu intenția explicită de a îmbunătăți performanța acestuia, care este investigată luând în considerare două cazuri de utilizare în IoT: un aeroport inteligent și o fabrică inteligentă. Părțile interesate (eng. stakeholders) în proiectarea acestui sistem sunt echipele de intervenție care trebuie să acționeze rapid pentru salvarea persoanelor în cazul detectării (și raționării) unor observații critice în mediul inteligent.

Conform (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007), după identificarea problemei de cercetare și precizarea obiectivelor tezei, următorul pas în procesul de Design Science este de a proiecta și dezvolta artefactul propus. Artefactul propus în această teză reprezintă un artefact *de instanțiere* (March & Smith, 1995; Hevner, March, Park, & Ram, 2004), întrucât sistemul de tip „pipeline” este implementat utilizând Apache Kafka[1], ca platformă distribuită pentru procesarea fluxului de date, și GraphDB[2], ca server de baze de date semantice.

[1]https://kafka.apache.org/
[2]http://graphdb.ontotext.com/

Pentru a obține o soluție adecvată, este nevoie ca sistemul de tip „pipeline” să fie dezvoltat în mod iterativ și să fie validat. Această cercetare utilizează două iterații anterioare construirii soluției finale. În fiecare iterație, este implementat și evaluat câte un prototip. Prima iterație constă în dezvoltarea unui sistem scalabil de procesare semantică a fluxului de date folosind un scenariu de tip aeroport inteligent. Analiza experimentală asociată acestei iterații ia în considerare diferite configurări ale clusterului Kafka. În procesul de evaluare a performanței prototipului asociat acestei iterații, folosim criteriul de scalabilitate. A doua iterație folosește ca input prototipul rezultat în cadrul primei iterații și utilizează un scenariu de tip fabrică inteligentă pentru a dezvolta un sistem cu latență scăzută. În cadrul celei de-a doua iterații implementăm diferite arhitecturi ale sistemului de tip „pipeline” pentru a investiga locul preferabil de a executa sarcina de adnotare semantică în vederea obținerii unor informații mai rapide din datele senzor generate în mediu. Iterația a doua reprezintă artefactul final al acestei cercetări.

5.2. Prezentarea generală a soluției propuse: un sistem de tip „pipeline” pentru procesarea semantică a fluxului de date

Componentele principale ale sistemului de tip „pipeline” sunt ilustrate în Figura 2. Sursele de date sunt reprezentate de diferiți senzori instalați în mediu. În această teză folosim senzori atașați pe corpul uman și senzori ambientali pentru a proiecta cazuri specifice de utilizare în IoT. Senzori audio geolocalizați sunt instalați în mediul unui aeroport inteligent, pe când senzori de puls, de localizare și de proximitate sunt utilizați pentru construirea unui ecosistem de fabrică inteligentă.

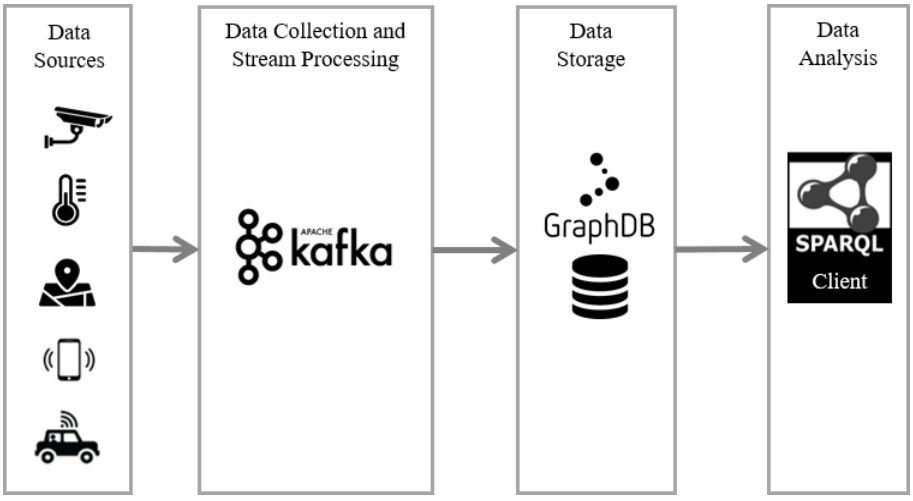

Figura 2. Sistemul de tip „pipeline” pentru procesarea semantică a fluxului de date senzor (Zălhan, Silaghi, & Buchmann, 2019)

Componenta de colectare și procesare a fluxului de date folosește un sistem distribuit de ingerare a datelor pentru a colecta în mod continuu datele de la senzori în vederea analizei lor ulterioare. În această teză folosim Apache Kafka (Narkhede, Shapira, & Palino, 2017) pentru a ingera fluxul de date senzor deoarece acesta ne permite să procesăm în timp real un debit mare de date și să stocăm aceste date într-un mod tolerant la erori.

Arhitectura de bază a Kafka este organizată în jurul câtorva termeni cheie: subiecte, producători, consumatori și brokeri. Unitatea de bază cu care funcționează Kafka este mesajul. Mesajele în Kafka sunt organizate pe subiecte (eng. topics). La rândul lor, subiectele sunt împărțite în partiții care sunt distribuite uniform în cadrul mai multor servere pentru a fi gestionate în paralel. Fiecare partiție poate fi replicată în cadrul mai multor servere pentru a garanta toleranță la erori. Astfel, partițiile sunt modalitatea prin care Kafka oferă scalabilitate și replicare.

Apache Kafka este cel mai performant sistem de mesagerie bazat pe modelul de „publish/subscribe", unde mai mulți producători postează mesaje într-un topic Kafka, iar mai mulți consumatori se abonează la un topic Kafka pentru a citi mesajele publicate anterior. Fiind un sistem distribuit, Kafka rulează într-un cluster, unde fiecare nod din cluster este numit broker. Intern, Kafka folosește Apache Zookeeper pentru coordonarea și managementul brokerilor.

Pentru a furniza descrieri concise și inteligibile care să poată fi manevrate de mașini, datele ingerate sunt adnotate semantic folosind ontologia SSN. În urmă, descrierile semantice rezultate sunt stocate într-o bază de date semantică pentru extragerea unor informații utile prin raționare, pentru executarea unor deducții logice prin interogări SPARQL sau pentru integrarea cu SI preexistente (ex., sistem de notificare). În această teză folosim serverul de baze de date semantice numit GraphDB pentru a persista fluxurile de date senzor adnotate semantic. Potrivit unui studiu recent (Bellini & Nesi, 2008) cu privire la evaluarea performanței serverelor de baze de date semantice, GraphDB depășește soluțiile existente având o performanță ridicată în ceea ce privește timpul de încărcare și indexare a depozitului de triplete. Timpul de încărcare al serverului de baze de date RDF influențează performanța sistemului de tip „pipeline", întrucât un timp de încărcare ridicat poate cauza întârzieri în livrarea datelor la timp.

Componenta de analiză a datelor este reprezentată de clientul SPARQL care accesează baza de date integrată formată din tripletele de date senzor și alte date preexistente. Fluxurile de date semantice stocate în baza de date de grafuri RDF sunt manipulate cu ajutorul limbajului de interogare SPARQL pentru a deduce noi cunoștințe și pentru a extrage informații utile.

5.3. Extensii propuse pentru ontologia SSN

Deoarece ontologia SSN identificată în literatură nu modelează o ierarhie de tipuri specifice de senzori, nici tipuri specifice de obiecte-de-interes, sau observații critice ale senzorilor, propunem câteva extensii ale ontologiei SSN existente, și anume:

- Ierarhizarea claselor de senzori, observații, obiecte-de-interes și platforme specifice.
- Legături directe între senzori și obiecte-de-interes, între senzori și observații critice, precum și relații între diferite obiecte-de-interes.

Pentru a depăși limitările ontologiei SSN existente, extindem sintaxa sa cu termeni noi, dintre care unii sunt independenți de scenariu, iar alții țintesc cazuri specifice din domeniul IoT. Baza de cunoștințe este îmbogățită cu un set de axiome care sunt definite cu terminologii standard precum RDFS, Schema.org și OWL.

Folosim proprietatea standard "rdfs:subClassOf" a vocabularului RDFS pentru a construi o ierarhie de clase senzor specifice, definind conceptul de senzor atașat pe corpul uman și senzor ambiental. De asemenea, introducem clase senzor specifice pentru a modela senzorii instalați în diferite medii inteligente.

Valorile senzorilor sunt comparate cu limite de prag pentru a diferenția valorile normale de valorile critice ce pot indica probleme serioase în mediul inteligent. În acest sens, creăm o taxonomie de observații, categorisind observațiile senzorilor în observații critice inferioare sau observații critice superioare.

Extindem clasa "sosa:FeatureOfInterest" pentru a crea o ierarhie de obiecte-de-interes. Definim noi concepte pentru a modela entități vii, precum actorii dintr-un scenariu IoT. În acest proces, folosim conceptul de Persoană furnizat de vocabularul Scheam.org. Similar, definim noi concepte pentru a modela entități pasive, și anume echipamente fixe, clădiri și medii. Pentru fiecare dintre aceste entități, furnizăm noi subclase pentru a susține scenarii IoT specifice.

Introducem noi relații între concepte cu ajutorul proprietății standard "rdfs:subPropertyOf". Fiecare proprietate introdusă are asociat un domeniu și un codomeniu. Stabilim legătura între un senzor și o observație critică (superioară sau inferioară). De asemenea, introducem noi proprietăți ce descriu relații între diferite obiecte-de-interes (active sau pasive) și senzori. Unele dintre aceste proprietăți sunt definite în ambele direcții cu ajutorul proprietăți "owl:inverseOf" furnizată de terminologia standard OWL pentru a facilita navigarea în graf.

Stabilim legături între diverse obiecte-de-interes cum ar fi între un echipament fix și mediul în care acesta este instalat, sau între un actor și mediul în care acesta acționează. De asemenea, definim o ierarhie de caracteristici observabile ale obiectelor-de-interes. Aceste caracteristici observabile leagă o observație a unui senzor de un rezultat.

5.4. Implementarea soluției

Configurările clusterului Kafka sunt realizate folosind Platforma Confluent[3] ce permite organizarea și gestionarea fluxurilor de date. Realizăm două configurări ale clusterului de Kafka și anume: (1) *un singur nod cu un broker unic*; (2) *un sigur nod cu mai mulți brokeri*. Aceste configurări vor fi testate ulterior în vederea alegerii configurării capabile să susțină sarcina de adnotare semantică a volumelor mari de date.

Pentru a identifica unde este preferabil de a executa sarcina de adnotare semantică a fluxului de date, am implementat două arhitecturi alternative ale sistemului de tip „pipeline", și anume:

(1) *Modelare semantică pe parte de consumator* – unde producătorii Kafka generează și publică fluxuri de date senzor în subiecte, în timp ce consumatorii Kafka se abonează la aceste subiecte, apoi citesc, adnotează fluxurile de date senzor și stochează fluxurile de date RDF în depozitul de triplete.

(2) Modelare semantică pe parte de producător – unde producătorii Kafka generează, adnotează fluxurile de date senzor și publică fluxurile de date RDF în subiecte, în timp ce consumatorii Kafka se abonează la aceste subiecte, apoi citesc și stochează fluxurile de date RDF în depozitul de triplete.

În ambele arhitecturi ale sistemului de tip „pipeline", fluxurile de date se conformează unei scheme formată din mai multe atribute care reprezintă identificatorul senzorului, tipul de senzor, identificatorul platformei care găzduiește senzorul, identificatorul fluxului de date, valoarea generată de senzor, precum și ștampila de timp asociată ce marchează momentul de timp când valoarea senzorului a fost generată în mediul inteligent.

Pentru a publica și abona fluxuri de date senzor în clusterul Kafka, am definit producători și consumatori Kafka utilizând clasele expuse de librăria kafka-python[4].

[3] https://www.confluent.io/product/confluent-platform
[4] http://github.com/dpkp/kafka-python

5.5. Validarea soluției

Performanța sistemului de tip „pipeline” este evaluată în funcție de mai multe metrici: debitul de date, scalabilitate și timp de execuție. Sistemul de ingerare al datelor care alimentează în mod continuu sistemul de tip „pipeline” cu date senzor, trebuie să fie capabil de a gestiona debit mare date. Mai mult, sistemul de tip „pipeline” trebuie să fie scalabil pentru a gestiona eficient debitul mare de date. Apache Kafka, ca și componentă de colectare și procesare a fluxurilor de date, a fost proiectat pentru a face față volumelor mari de date. Kafka prezintă scalabilitate orizontală atât pentru producătorii, cât și pentru consumatorii de mesaje. De asemenea, suntem interesați în a determina timpul necesar pentru a genera un mesaj în cele două arhitecturi alternative ale sistemului.

6. Cazuri de utilizare IoT în mediile inteligente

În acest capitol prezentăm două cazuri de utilizare IoT, pentru a ilustra modul în care poate fi folosit în scenarii reale sistemul de tip „pipeline” propus în Capitolul 5. Cele două cazuri fac parte din categorii diferite de cazuri de utilizare în IoT. Aplicația de aeroport inteligent face parte din categoria cazurilor de utilizare guvernamentale, pe când aplicația de fabrică inteligentă face parte din categoria cazurilor de utilizare industriale. Pentru fiecare dintre aceste aplicații în IoT prezentăm mai întâi scenariul, apoi câteva decizii de proiectare a mediului inteligent, iar pe urmă discutăm rezultatele experimentale obținute.

6.1. Aeroport inteligent

Pentru a preveni viitoare atacuri teroriste asupra unui aeroport, infrastructura unui aeroport inteligent este dotată cu senzori audio și tehnologii moderne precum tehnologia de recunoaștere automată a vorbirii (eng. Automatic Speech Recognition - ASR) și tehnologii de geo-localizare. Rolul tehnologiei ASR în contextul unui aeroport inteligent este de a detecta conversații suspicioase între pasageri și a notifica operatorii de securitate să acționeze în conformitate. Componenta de recunoaștere vocală nu este implementată în acest studiu de cercetare, întrucât ne focalizăm pe construirea unui arhitecturi care să susțină adnotarea semantică a fluxului de date senzor, precum și integrarea acestor date cu date preexistente. Folosim experiența anterioară (Zălhan, Stan, Teodorescu, Saupe, & Duma, 2016) cu privire la construirea unui sistem ASR iar această componentă face parte din dezvoltările viitoare ale sistemului de tip „pipeline”.

Pentru a simula un flux de date continuu generat de către senzorii audio geolocalizați, am creat producători și consumatori Kafka care scriu și citesc fluxuri de date în clusterul Kafka. Mesajele publicate de producători sunt scrise într-un subiect. În Figura 3, prezentăm schema JSON unui flux brut de date și descrierea semantică asociată din perspectiva observației generate de senzorul audio. Pentru că ontologia SSN nu modelează unități de măsură, nici comportamente dinamice, cum ar fi localizarea unui senzor în mediul inteligent, în procesul de adnotare semantică folosim vocabularul „Quantities, Units, Dimensions and Data Types” (QUDT)[5] pentru a modela unități

[5] http://www.qudt.org/

acustice, dar și vocabulare geospațiale precum GeoSPARQL[6] pentru modelarea locației senzorului audio în mediu.

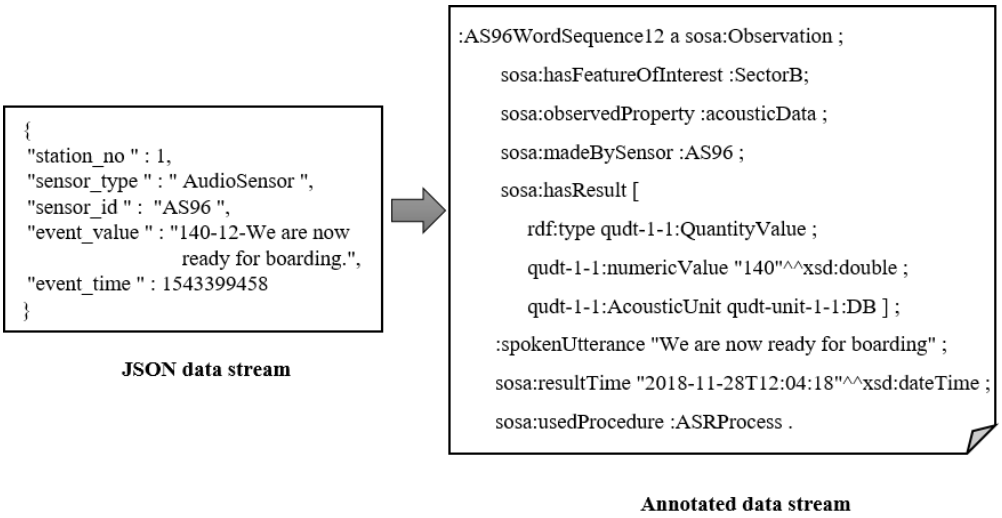

Figura 3. Descrierea unui observații capturate de un senzor audio (Zălhan, Silaghi, și Buchmann, 2019)

De îndată ce datele adnotate semantic sunt persistate în baza de grafuri RDF, aplicăm interogări SPARQL asupra acestora în vederea extragerii de informații utile, a generării de noi cunoștințe din cele existente sau a integrării lor cu date preexistente. Cu ajutorul unei interogări mai complexe, sistemul notifică operatorii de securitate responsabili de un anumit sector în aeroport în care a fost detectată o observație critică superioară, prin trimiterea unui mesaj pe telefon. În definirea acestei interogări, integrăm baza de date tradițională (îmbogățită semantic) ce conține informații despre angajații aeroportului și zonele în care aceștia lucrează, cu baza de grafuri RDF ce conține descrieri semantice ale datelor senzor.

Folosim ecosistemul de tip aeroport inteligent pentru a evalua performanța sistemului de tip „pipeline” prin realizarea a două configurări ale clusterului Kafka diferite: (1) un singur nod cu un broker unic; (2) un sigur nod cu mai mulți brokeri. Rezultatele experimentale din Figura 4 arată că, în al doilea scenariu Kafka, clusterul Kafka reușește să ingereze volume mari de date distribuind uniform publicarea datelor în cadrul mai multor brokeri. De asemenea, configurarea distribuită a clusterului Kafka reușește să susțină adnotarea semantică a volumelor mari de date senzor, prin intermediul multiplelor instanțe de consumator care citesc în mod concurent mesajele dintr-un subiect.

[6]https://www.opengeospatial.org/standards/geosparql

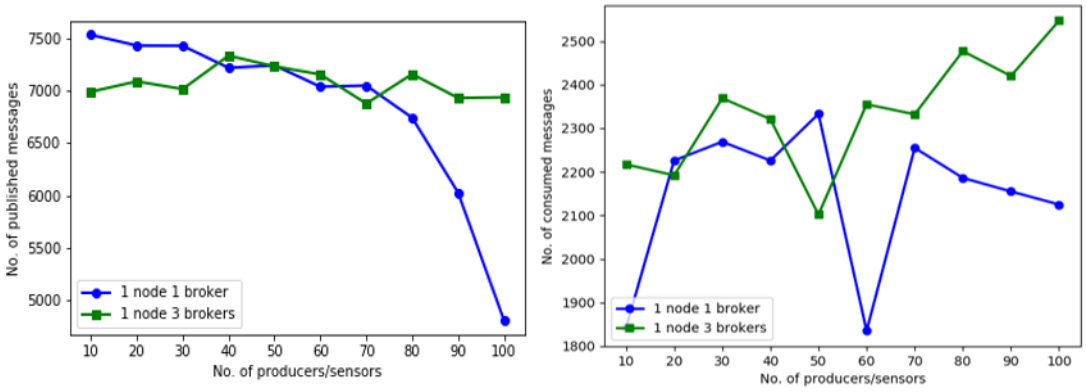

Figura 4. Numărul de producători vs. numărul de mesaje publicate sau consumate în ambele configurări Kafka

6.2. Fabrică inteligentă

Ecosistemul de fabrică inteligentă este centrat pe siguranța muncitorilor prin monitorizarea parametrilor de sănătate în timp real cu ajutorul unor senzori de puls, de localizare și de proximitate. Crearea unui loc de muncă mai sigur este realizată prin implicarea unor echipe de intervenții în caz de incidente sau dezastre naturale pentru prevenirea leziunilor și a deceselor angajaților fabricii. Senzorii de puls și de localizare sunt atașați pe corpul uman pentru a monitoriza bătăile inimii și coordonatele spațiale ale unui angajat. Senzori de proximitate sunt instalați pe mașinăriile din fabrică pentru a detecta o prezență umană de-a lungul unei distanțe, cum ar fi un muncitor deplasându-se prea aproape de o mașină grea.

Pentru a simula un flux de date continuu generate de către senzorii de puls, de localizare și de proximitate, am creat producători și consumatori Kafka care scriu și citesc fluxuri de date în clusterul Kafka. Mesajele publicate de producători sunt scrise în diferite subiecte. Schema de adnotare a fluxurilor de date brute, original generate în formatul JSON, este similară cu cea folosită în cazul senzorilor audio din aeroportul inteligent. Folosim vocabularul QUDT pentru modelarea unităților de măsură asociate valorilor observate (bătăile inimii pentru senzorul de puls, repsectiv distanța în centimetri pentru senzorul de proximitate), precum și vocabularul geospațial numit WGS84 Geo Positioning[7] pentru modelarea locației dinamice a unui muncitor în fabrică.

De îndată ce datele adnotate semantic sunt persistate în baza de grafuri RDF, aplicăm interogări SPARQL asupra acestora în vederea extragerii de informații utile sau a generării de noi cunoștințe

[7]https://www.w3.org/2003/01/geo/wgs84_pos

din cele existente. Creăm interogări complexe în vederea obținerii coordonatelor spațiale asociate unui muncitor a cărui senzor de puls a detectat o valoare critică inferioară. De asemenea, executăm interogări SPARQL pentru identificarea celui mai apropiat muncitor, ce se află în proximitatea unei mașinării grele.

Folosim ecosistemul de tip fabrică inteligentă pentru a evalua performanța sistemului de tip „pipeline" prin implementarea a două arhitecturi alternative, și anume: modelare semantică pe parte de consumator (S1); respectiv modelare semantică pe parte de producător (S2). În arhitectura (S2), fluxurile brute de date senzor sunt procesate semantic la marginea rețelei, aproape de sursa de date.

Tabel 1. Numărul de mesaje adnotate și tripletele RDF corespunzătoare în ambele arhitecturi

Producători	S1		S2		Multiplicarea debitului de date
	Mesaje	Triplete RDF	Mesaje	Triplete RDF	
10	529	8993	5703	96951	10,78
20	558	9486	6253	106301	11,21
30	696	11832	6207	105519	8,92
40	579	9843	6023	102391	10,40
50	357	6069	6043	102731	16,93
60	652	11084	5714	97138	8,76
70	707	12019	5902	100334	8,35
80	851	14467	5904	100368	6,94
90	751	12767	5911	100487	7,87
100	705	11985	6091	103547	8,64

Tabelul 1 centralizează numărul de mesaje adnotate și numărul de triplete RDF corespunzătoare, din ambele arhitecturi (S1 și S2). În arhitectura S2, când procesarea semantică este realizată la periferia rețelei de senzori, sunt generate chiar de 10 ori mai multe mesaje adnotate (implicit, și numărul de triplete RDF corespunzătoare este aproximativ de 10 ori mai mare) decât în arhitectura S1. În mediul de tip „semantic edge", datele senzor nu irosesc timp traversând întreaga conductă de date ci sarcina de adnotare semantică este executată local.

7. Concluzii și direcții viitoare de cercetare

Teza actuală propune un sistem de tip „pipeline” pentru procesarea semantică a fluxului de date senzor, oferind două aplicații interesante din domeniul IoT pentru a-i evalua performanța. Arhitectura sistemului propus conține mai multe componente care acoperă toate aspectele unui lanț integral de prelucrare a datelor, și anume: colectarea datelor, adnotare semantică, stocarea tripletelor RDF și executarea unor interogări SPARQL asupra fluxurilor de date senzor. Pe întreg parcursul cercetării ne-am direcționat eforturile pentru a atinge obiectivele propuse.

Pe baza unei revizuiri a literaturii de specialitate din domeniul procesării semantice a fluxului de date, am analizat abordările adoptate de diferiți autori pentru procesarea în timp real a volumelor mari de date generate în mod continuu de surse eterogene de date și extragerea de informații utile din acestea. Mai mult, am identificat limitările sistemelor existente privid procesarea semantică a fluxurilor de date, astfel că, niciun sistem existent nu propuse o soluție de edge computing pentru o analiză mai rapidă și mai eficientă a fluxului de date. De asemenea, ontologia SSN identificată în literatură nu modelează tipuri de senzori, unități de măsură, sau comportamente dinamice precum geolocalizare. Pentru a depăși aceste limitări, extindem ontologia SSN cu termeni noi ce descriu ierarhii de senzori, de observații critice, de obiecte-de-interes cu caracteristici observabile, precum și relații între senzori și observații critice, între senzori și obiecte-de-interes, relații între diferite obiecte-de-interes.

Soluția middleware propusă în această teză folosește un alt sistem distribuit de mesagerie numit Apache Kafka, deoarece acesta este capabil de a procesa un debit mai mare de date decât alte sisteme de mesagerie similare, iar prin mecanismul de partiționare incorporat permite procesarea paralelă a datelor de-a lungul mai multor mașini de calcul din cluster. Un alt aspect care diferențiază soluția propusă de alte soluții existente este abordarea combinată de adnotare semantică, îmbinând ontologia SSN cu alte vocabulare.

Pentru a obține interoperabilitate semantică cu SI care se bazează pe fluxurile de date adnotate, integrăm semantic baza de grafuri de date senzor cu baza de date preexistente pentru a susține dezvoltarea unui sistem semantic hibrid pentru gestionarea incidentelor. În acest proces, hibridizăm ontologia SSN existentă cu termeni din alte vocabulare pentru a modela informații

despre persoanele membre din echipele de intervenție ce își desfășoară activitatea în mediile inteligente.

Ca direcție viitoare de cercetare, intenționăm să îmbunătățim performanța sistemului propus prin configurarea unei soluții distribuite Kafka cu mai multe noduri și mai mulți brokeri. Această configurare a clusterului Kafka este de dorit pentru a crea o infrastructură de calcul de înaltă performanță necesară analizei volumelor mari de date.

Referințe

Bellini, P., & Nesi, P. (2008). Performance assessment of rdf graph databases for smart city services. (Elsevier, Ed.) *Journal of Visual Languages and Computing, 45*, 24-38.

Cisco. (2018, February). *Cisco Edge-to-Enterprise IoT Analytics for Electric Utilities Solution Overview*. Preluat pe March 25, 2020, de pe Cisco: https://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/big-data/solution-overview-c22-740248.html

Domingue, J., Fensel, D., & Hendler, J. (2011). *Handbook of Semantic Web Technologies.* Springer.

Firouzi, F., Chakrabarty, K., & Nassif, S. (2020). *Intelligent Internet of Things: from Device to Fog and Cloud.* Springer.

Govindan, K., & Azad, A. P. (2015). End-to-end service assurance in IoT MQTT-SN. *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)* (pg. 290-296). IEEE Press.

Haller, A., Janowicz, K., Cox, S., Le Phuoc, D., Taylor, K., & Lefrançois, M. (2017, October 19). *Semantic Sensor Network Ontology*. Preluat de pe W3C Recommendation: https://www.w3.org/TR/vocab-ssn/

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly, 28*(1), 75-105.

Janowicz, K., Haller, A., Cox, S. J., Le Phuoc, D., & Lefrançois, M. (2019). SOSA: A lightweight ontology for sensors, observations, samples, and actuators. *Journal of Web Semantics, 56*, 1-10.

Jerry, M. (2019, March). *SOA vs. EDA: Is Not Life Simply a Series of Events?* Preluat pe March 5, 2020, de pe Confluent: https://www.confluent.io/blog/soa-vs-eda-is-not-life-simply-a-series-of-events

Klyne, G., Carroll, J. J., & McBride, B. (2009). *Resource description framework (rdf): concepts and abstract syntax, 2004*. Preluat de pe http://www. w3. org/TR/2004/REC-rdf-concepts-20040210

Kreps, J. (2014, July 2). *Questioning the Lambda Architecture*. Preluat pe August 6, 2019, de pe O'Reilly: https://www.oreilly.com/ideas/questioning-the-lambda-architecture

Le Phuoc, D., & Hauswirth, M. (2019). Semantic Stream Processing. În S. Sakr, & A. Y. Zomaya, *Encyclopedia of Big Data Technologies.* Springer. doi:10.1007/978-3-319-63962-8_287-1

Le-Phuoc, D., Parreira, J. X., & Hauswirth, M. (2012). Linked Stream Data Processing. *Reasoning Web. Semantic Technologies for Advanced Query Answering*, (pg. 245-289). Vienna. doi:10.1007/978-3-642-33158-9_7

Li, Y., & Zhong, N. (2004). Web mining model and its applications for information gathering. *Knowledge-Based Systems, 17*(5), 207-217.

Liu, X., Dastjerdi, A., & Buyya, R. (2016). Stream processing in IoT: Foundations, state-of-the-art, and future directions. În R. Buyya, & A. Dastjerdi, *Internet of Things: Principles and paradigms* (pg. 145-161). Elsevier.

March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision support systems, 15*(4), 251-266.

Marz, N., & Warren, J. (2015). *Big Data: Principles and best practices of scalable real-time data systems.* New York: Manning Publications Co.

Motik, B., Patel-Schneider, P. F., Parsia, B., Bock, C., Fokoue, A., Haase, P., . . . Sattler, U. (2009). OWL 2 web ontology language: Structural specification and functional-style syntax. *W3C recommendation, 27*(65), 159.

Narkhede, N., Shapira, G., & Palino, T. (2017). *Kafka: the definitive guide: real-time data and stream processing at scale.* O'Reilly Media, Inc.

Noura, M., Atiquzzaman, M., & Gaedke, M. (2019). Interoperability in Internet of Things: Taxonomies and Open Challenges. *Mobile Networks and Applications, 24*(3), 796-809. doi:10.1007/s11036-018-1089-9

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems, 24*(3), 45-77. doi:10.2753/MIS0742-1222240302

Stack, T. (2018, February). *Internet of Things (IoT) Data Continues to Explode Exponentially. Who Is Using That Data and How?* Preluat pe April 25, 2020, de pe Cisco Blogs: https://blogs.cisco.com/datacenter/internet-of-things-iot-data-continues-to-explode-exponentially-who-is-using-that-data-and-how#comments

Stanford-Clark, A., & Truong, H. L. (2013). MQTT for sensor networks (MQTT-SN) protocol specification. *International business machines (IBM) Corporation version, 1*, 2.

Stonebraker, M., Cetintemel, U., & Zdonik, S. (2005). The 8 requirements of real-time stream processing. *ACM Sigmod Record, 34*(4), 42-47.

Vadivel, V., & Subramanian, S. (2017). Semantic Technologies for IoT. În B. Tripathy, & J. Anuradha, *Internet of Things (IoT): Technologies, Applications, Challenges, and Solutions* (pg. 265-294). CRC Press.

Wieringa, R. J. (2014). *Design science methodology for information systems and software engineering.* Springer.

Zălhan, P.-G. (2019). Transforming Big Data into knowledge using semantic stream processing technology: challenges and early progress. *Proceedings of the 18th International Conference on INFORMATICS in ECONOMY (IE 2019) Education Research & Business Technologies*, (pg. 365-370). București.

Zălhan, P.-G., Silaghi, G. C., & Buchmann, R. A. (2019). Marrying Big Data with Smart Data in Sensor Stream Processing. *A. Siarheyeva, C. Barry, M. Lang, H. Linger, & C. Schneider (Eds.), Information Systems Development: Information Systems Beyond 2020 (ISD2019 Proceedings).* Toulon, France. Preluat de pe https://aisel.aisnet.org/isd2014/proceedings2019/ManagingISD/8/

Zălhan, P.-G., Stan, A., Teodorescu, L.-R., Saupe, A.-B., & Duma, M. (2016). A Kaldi-based ASR Solution for the Romanian Judicial System. *International Conferece on INFORMATICS in ECONOMY (IE 2016): Education, Research & Business Technologies*, (pg. 191-197). Cluj-Napoca.