UNIVERSITATEA BABEŞ-BOLYAI
FACULTATEA DE ŞTIINȚE ECONOMICE ŞI GESTIUNEA AFACERILOR
DEPARTAMENTUL DE INFORMATICĂ ECONOMICĂ

Teză de dotorat:

MODELAREA PROCESELOR DECIZIONALE ECONOMICE FOLOSING INSTRUMENTE INTELIGENTE DE MINING

- rezumat-

Coordonator ştiinţific:
Prof. univ. dr. Nicolae **TOMAI**

Student doctorand:
Cristina-Claudia **DOLEAN**

Cluj-Napoca, 2013

Cuprinsul tezei

Cuprins

Abstract

Sistemele informatice de tip process-aware (de exemplu: Sistemele de Management al Fluxului de Activități, Sistemele de Management ale Proceselor de Afaceri) generează loguri de evenimente care reprezintă acțiunile făcute de către utilizatori în cadrul sistemelor informatice. Fiecare log stochează informații cu privire la resursa care a executat activitatea, timpul când activitatea a fost începută/finalizată sau elemente de date înregistrate în cadrul evenimentului (de exemplu: numele clientului unei bănci). Asupra acestor loguri pot fi aplicate tehnici de process mining şi pot fi descoperite modele care evidențiază fluxul activităților. Aşadar scopul analizei proceselor prin process mining este de a extrage informații despre procesul ascuns în aceste loguri (de evenimente). Cel mai cunoscut şi simplu algoritm care oferă o perspectivă a fluxului de activități a unui proces este algoritmul α. Există mai mulți astfel de algoritmi a căror scop este de a extrage modele din loguri, cum ar fi: Heuristics Miner, Genetic Miner, Fuzzy Miner şi alții.

Aşa cum am menționat mai devreme, literatura de specialitate oferă multe informații privind aspecte legate de fluxul activităților unui process, ignorând perspectiva fluxului de date sau cel mult integrând-o cu cea a fluxului de activități. Fluxul de activități prezintă ordinea activităților în cadrul proceselor, dar ignoră fluxul datelor. Unele date pot să nu fie disponibile la timp pentru executarea unei anumite activități şi fluxul de lucru îşi opreşte execuția. Aceasta problemă a fost deja cercetată. Procesul poate fi îmbunătățit analizând doar fluxul de date sau ambele modele: modelul fluxului de date şi modelul fluxului de activități.

Problema identificată de noi în cadrul cercetărilor anterioare este că nicio metodă sau tehnică propusă nu foloseşte ca punct de pornire analiza datelor cu privire la evenimentele unui log (principala componentă a process mining-ului). Literatura oferă o serie de analize bazate pe fluxul datelor unui proces. Cu privire la analiza fluxului de date a unui proces, cercetătorii au încercat să ofere o vizualizare a fluxului de date de sine stătătoare sau au combinat fluxul de activități cu fluxul de date. În acest sens, au fost propuse metode de validate a datelor şi au fost identificate câteva erori ale fluxului de date care pot avea loc în cadrul procesului. Există, totuşi, un model care evidențiază mişcarea datelor în cadrul unui proces: Product Data Model (PDM). Neajunsul acestei abordări este că nu propune nicio metodă automată de extragere a PDM-ului (doar una manuală). Teza de față încearcă să umple acest gol prin propunerea unor metode automate sau semi-automate care furnizează o vizualizare a procesului bazată pe date.

Propunem trei algoritmi de mining care se axează pe date pentru extragerea fluxului de date a unui proces. Dar logurile trebuie să respecte un anumit format. În acest sens, am definit o extensie şi am implementat două instrumente de conversie. Pentru validare am utilizat loguri generate de către un sistem de tip ERP (*Navision*), respectiv de către un sistem de management al fluxului de activități (*YAWL*). În fiecare caz am comparat rezultatele obținute folosind algoritmii de mining bazați pe date cu modele de proces care subliniază perspectiva fluxului de activități.

Keywords: Product Data Model, flux de lucru, fluxul activităților, fluxul de date, algoritimi de mining, PAIS, elemente de date de intrare, elemente de date de ieşire, operații

Introducere

Domeniul de process mining oferă o serie de tehnici care analizează modelele de procese ascunse în "amprentele" lăsate de către utilizatori în cadrul sistemelor informatice. Aceste urme sunt numite loguri de evenimente şi sistemele informatice capabile să genereze loguri de evenimente prin integrarea aplicațiilor, resurselor şi proceselor sunt numite Sisteme de tip Process-Aware (PAIS). Există trei tipuri de process mining: descoperire, conformitate şi îmbunătățire. Tehnicile de descoperire se referă la extragerea modelelor de procese din loguri de evenimente. Pe lângă descoperire, proces mining-ul analizează abaterile între comportamentul dorit al unui proces şi cel real al acestuia. Aici vorbim despre conformitate. Ultima componentă a process mining-ului se referă la procesul de îmbunătățire pe baza informațiilor existente în logurile de evenimente.

În ziua de azi, majoritatea companiilor folosesc sisteme informatice pentru gestionarea propriilor afaceri (de exemplu sisteme de tip ERP). Sisteme de tip ERP sunt concepute pentru a administra toate datele conexe logisticii, resurselor umane, producției şi respectiv, vânzării. Mai mult decât atât sistemele de tip ERP sunt capabile a partaja date între diferite departamente. În general, angajații unei companie cunosc procesele interne ale companiei. Dar după executarea unui anumit număr de activități, întrabarea care rezultă este legată de datele necesare pentru a continua procesul. În plus, fiind într-un anumit punct al execuției procesului (după executarea unui anumit număr de activități), întrebările adresate de către angajati sunt "care sunt datele disponibile şi pot/trebuie să fie folosite în activitățile viitoare ale procesului"; sau "care sunt datele de care mai avem nevoie pentru a executa o anumită activitate". Perspectiva fluxului de activități nu are capacitatea de a oferi aceste răspunsuri. Fiind într-un anumit punct al execuției unui process, poate doar arăta care sunt activitățile care pot/trebuie să fie executate, dar nu poate asigura executarea activităților.

Motivația acestei teze este de a oferi mai multe detalii cu privire la perspectiva de date a modelelor de procese. De ce vrem să analizăm aceste aspecte? *În primul rând*, tehnicile de process mining existente se concentrează asupra aspectelor legate de fluxul activităților unui proces și perspectiva fluxului de date este ignorată sau cel mai integrată cu fluxul de activități. Fluxul de activități prezintă ordinea activităților procesului, dar ignoră mișcările de date din cadrul acestuia. *În al doilea rând*, perspectiva fluxului de date poate aduce îmbunătățiri procesului (de exemplu, dacă modelul fluxului de date scoate în evidență o operație cu frecvență mică putem verifica dacă această operație este într-adevăr executată câteva ori sau reprezintă o deviere de la comportamentul dorit al procesului). *În al treilea rând*, presupunând că executăm un proces și suntem într-un anumit punct din execuția acestuia, putem identifica, pe baza datelor cunoscute până stadiul actual al procesului, care sunt operațiile care pot fi executate în continuare. Mai mult, un model al fluxului de date al unui proces ne arată care sunt datele suplimentare de care avem nevoie pentru a executa o anumită operație. *În al patrulea rând*, niciuna din metodele și tehnicile propuse nu au ca punct de pornire analiza datelor cu privire la evenimentele unui log (principala componentă a process mining-ului). Există o abordare care oferădetalii legate de perspectiva fluxului de date a unui proces, dar nicio metodă automată nu este propusă pentru generarea acesteia (doar una manuală). Mai mult decât atât, o analiză a dependențelor datelor, de asemenea există, dar nici aceasta nu oferă o extragere automată a fluxului de date. În general,

cercetătorii au considerat fluxul de date deja creat de către experți sau l-au extras pe baza analizei semantice.

Figura 1 oferă o privire de ansamblu asupra structurii tezei. Aceasta evidențiază principalele aspecte ale fiecărui capitol. Primele trei capitole oferă o analiză a cercetărilor efectuate în domeniile adiacente modelării (Proiectarea Proceselor de Afaceri, Proiectarea Sistemelor şi Process Mining).

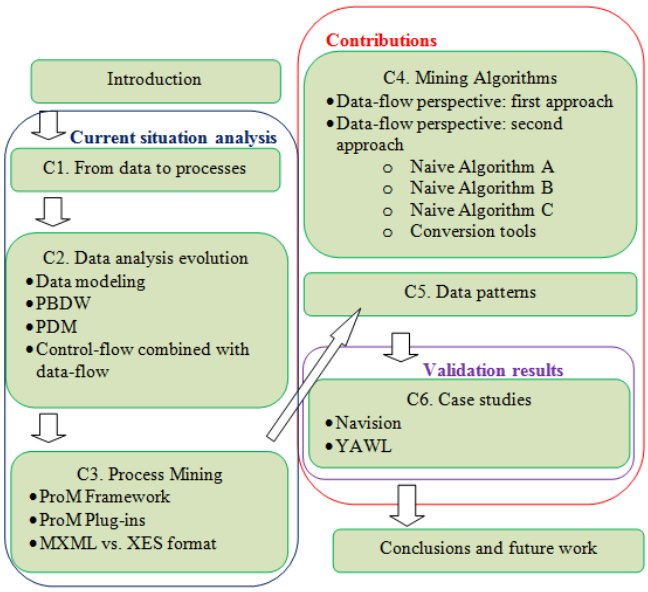

Figura 1 Imaginea de ansamblu a tezei

Primul capitol descrie modul în care sistemele informatice s-au schimbat, porinind de la sisteme orientate pe date spre cele orientate pe procese. Acest lucru a dus la apariția Sistemelor de Management al Fluxului de Lucru (WfMS). În primul rând, vom analiza dimensiunile fluxurilor de lucru. Apoi, vom vorbi despre şabloane ale fluxului de lucru, cu scopul de a sublinia importanța datelor în cadrul execuției proceselor. În cele din urmă, ne vom axa pe perspectiva fluxului de activități și cea a fluxului de date.

Capitolul 2 prezintă stadiul actual al cercetării privind metodele de analiză a datelor existente. Vom începe prin a prezenta tehnici de bază de modelare a datelor, cum ar fi Diagrama Entitate Relație (ERD) [12]. Apoi, vom vorbi despre unele tehnici care se concentrează pe perspectiva de flux a activităților precum Diagrama de Activități UML. Literatura de specialitate arată existența unor tehnici mixte care reunesc perspectiva datelor cu cea a fluxului de activități (de exemplu Diagrama de Date a Procesului). Acest capitol introduce, de asemenea, una dintre principalele noțiuni ale acestei tezei de doctorat: modelul Product Data Model (PDM), definit în [33].

Capitolul 3 face o introducere în domeniul numit process mining. Acesta analizează logurile de evenimente ale unor procese reale și le utilizează, în scopul extragerii de modele. O serie de formate standard, cum ar fi MXML (Mining Extensible Markup Language) și XES (eXtensible Event Stream) au fost definite datorită faptului că fiecare sistem informatic generează loguri de evenimente în propriul său mod. Prin urmare, a fost dezvoltat un instrument de mining care suportă o serie de tehnici unificate de către un format standardizat (MXML sau XES) și care oferă suport în analiza logurilor de evenimente– Platforma ProM.

Capitolele patru, cinci şi şase subliniază contribuțiile tezei. Capitolul 4 propune o serie de algoritmi implementați și validarea acestora cu ajutorul unor loguri de evenimente sintetice, în timp ce capitolul 6 validează ipotezele noastre din cadrul Capitolului 4, prin aplicarea algoritmilor naivi asupra unor loguri de evenimente. În continuare va fi făcută o scurtă prezentare a acestor capitole.

Capitolul 4 aduce principala contribuție formală a acestei teze și se axează pe abordările noastre legate de descoperirea perspectivei fluxului de date. Aici, vom introduce algoritmii de mining care produc perspectiva de flux a datelor unui proces. Vom începe cu analiza logurilor generate de către un sistem informatic de tip decision-aware (DAIS), iar apoi vom vorbi despre cei trei algoritmi naivi care generează perspectiva de flux a date a unui proces. În acest capitol sunt descrise funcționalitățile fiecarui algoritm. *Capitolul 5* propune o serie de şabloane de date (de exemplu, şabloane de date de bază, şabloane complexe de date și şabloane de valori ale datelor).

Capitolul 6 validează algoritmii de mining propuşi în Capitolul 4. Validarea acestora se bazează pe două studii de caz. Pentru primul studiu de caz am folosit logurile de evenimente produse de un sistem ERP (*Navision*), în timp ce pentru al doilea studiu de caz am folosit logurile de evenimente generate de un PAIS (*YAWL*). În scopul de a converti înregistrările bazei de date relaționale din Navision sau logurile de evenimente din YAWL către formatul dorit am folosit XESame 1.3 şi încă două instrumente de conversie. Având în vedere logurile de evenimente generate de YAWL, extragerea elementelor PDM-ului a fost posibilă prin analiza evenimentelor de tip *start* şi *complete*. Apoi, pentru fiecare studiu de caz am aplicat primii doi algoritmi naivi și am comparat rezultatele. Mai mult decât atât, am comparat modelele noastre cu un model care evidențiază perspectiva fluxului de activități, precum Alpha Miner.

În cele din urmă, ultimul capitol conclude teza și propune cercetările viitoare.

1. De la date la procese

Primul capitol prezintă natura sistemelor informatice începând cu cele bazate pe date din cadrul anilor '70 pâna la cele recente bazate pe procese. Apoi am făcut trecerea către automatizarea proceselor; aici am vorbit despre Sisteme de Management al Fluxului de Lucru (WfMSs). Am discutat aspecte legate de două instrumente comerciale furnizate de Software AG, respectiv de TIBCO. De asemenea am amintit un Sistem de Management al Fluxului de Lucru de tip open-source: în acest sens am descris de sistemul YAWL. Unul dintre cele mai importante aspecte tratate în acest capitol este perspectiva de date a unui proces.

Interesul pentru procese și reproiectarea proceselor de afaceri a apărut la începutul anilor ’90. Aici vorbim despre abordări bazate pe procese. Hammer și Champy [19] au definit procesul ca fiind "*o colecție de activități care au una sau mai multe tipuri de intrări și creează o ieșire, care aduce valoare clientului*". În cazul în care procesul este automatizat, prin specificarea informațiilor și sarcinilor necesare pentru fiecare activitate cu privire la un set de reguli, acesta devine flux de lucru. Acest lucru a condus la dezvoltarea Sistemelor de Management al Fluxului de Lucru (WfMS). „*Un Sistem de Management al Fluxului de Lucru (WfMS) este un instrument software generic, care permite definirea, executarea, înregistrarea și controlul fluxurilor de*

lucru" [11]. Pe piața există mai multe Sisteme de Management a Fluxurilor de Lucru (de exemplu Staffware, Aris, Jboss, WjMOpen, jBMP etc.). Platforma ARIS oferă o serie de instrumente care ajuta la procesul de analiză (de exemplu, ARIS Business Designer - pentru modelare, ARIS Business Architect - pentru rapoarte, ARIS Simulation - pentru simularea modelul etc.). ARIS Business Server stochează un server central de baze de date (de exemplu Oracle) utilizat pentru administrarea datelor. TIBCO Staffware oferă de peste 15 ani o soluție completă de Management a Proceselor de Afaceri pentru organizații. TIBCO a preluat Staffware din 2004, prin urmare, Suita TIBCO Staffware Process este rezultatul acestei uniuni. Aceasta asigură integrarea cu infrastructura IT existentă și aplicațiile unei companii. YAWL (Yet Another Workflow Language) este un sistem de flux de lucru bazat pe rețele Petri [25] care acceptă executarea de fluxuri de lucru. În acest sens, sunt utilizate mai multe şabloane ale fluxului de lucru [23]. YAWL este un instrument informatic de tip open-source dezvoltat de un grup de cercetători de la Universitatea Tehnică din Eindhoven, Olanda și Universitatea Tehnică din Queensland, Australia și permite modelarea activităților din cadrul unui proces. De asemenea, acesta oferă posibilitatea de a popula activitățile cu variabile și de stabili resursele care execută activitățile. Managementul Fluxurilor de lucru evidențiază fluxul activităților din cadrul proceselor şi se bazează pe rețele Petri, în timp ce Managementul Proceselor de Afaceri adaugă la această coordonarea proceselor.

Workflow Management Coalition propune trei dimensiuni ale fluxurilor de lucru. Dimensiunea fluxurilor de activități/procese oferă o vedere a întregului proces analizând activitățile implicate, în timp ce perspectiva de resurse distribuie activitățile unor resurse specifice. Dimensiunea de caz prezintă informații legate de o singură execuție unui anumit flux de lucru (process de afaceri). Perspectiva fluxului de activități analizează ordinea acestora, care activitate trebuie executată și când. Activitățile sunt executate de către resurse cu anumite roluri. Dimensiunea caz este formată dintr-o serie de instanțe de proces (o serie de activități).

În ultimul deceniu, o importanță majoră a fost acordată perspectivei de flux a activităților din cadrul proceselor, iar analiza perspectivei fluxului de date a fost aproape ignorată. Dar datele sunt necesare pentru a executa activitățile dintr-un (model de) proces.

Figura 2 prezintă, parțial, activitățile din cadrul unui hotel: de la înregistrarea clientului la operațiunea de check-out a acestiuia. Ne vom concentra pe primele două activități ale procesului. Dacă urmărim fluxul activităților procesului observăm ordinea de execuție a activităților (de exemplu *Înregistrare Client*, *Alocarea Camerei* etc.). Fiecare activitate necesită date (elemente de date de intrare), pentru a putea fi executată și la rândul său, produce date (elemente de date de ieșire). În exemplul nostru execuția activității *înregistrareClient* este posibilă dacă toate câmpurile sunt completate de către recepționistul hotelului (nume, prenume, titlu, stradă, oraș, cod poștal, țară, telefon și e-mail). Aceste aspecte nu sunt evidențiate de perspectiva fluxului de activități.

Mai departe aceste date sunt transferate următoarei activități (*alocarea camerei*). Chiar şi ultima activitate a procesului (*Inregistrare check-out*), are nevoie de date cu privire la client (elemente de date din cadrul primei activități). A doua activitate, de asemenea, are nevoie de date pentru a putea fi executată. La sfârşit toate datele sunt disponibile (numărul camerei, tipul camerei, numărul de nopți, numărul de adulți, numărul de copii și note).

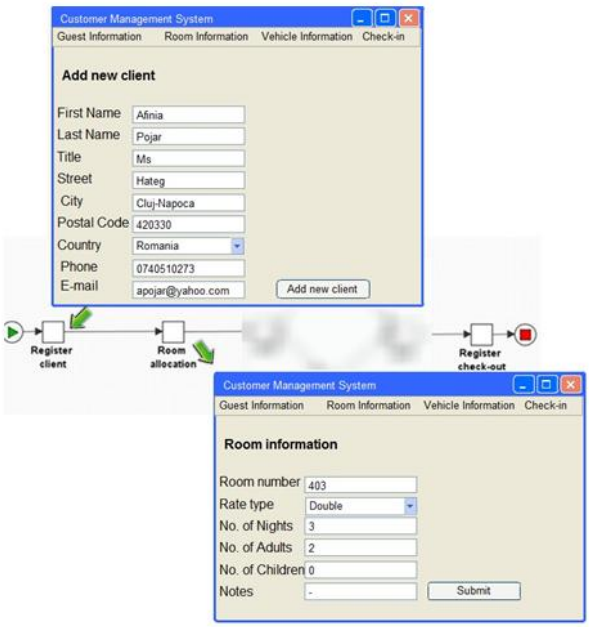

Figura 2 Datele din spatele activităților

Astfel datele se deplasează în cadrul unui proces, de la o activitate la alta. Elementele de date produse de o activitate poate fi utilizate de către celelate activități ale procesul. Prin urmare, dacă ne referim la procesele care conțin o serie de activități care consumă și produc mai multe elemente de date există o mișcare complexă a date în interiorul procesului.

În continuare vom introduce informații referitoare la şabloanele fluxurilor de lucru. Există mai multe tipuri de şabloane de fluxuri de lucru (de exemplu, şabloane ale fluxurilor de activități, şabloane ale fluxurilor de date, şabloane ale resurse, şabloane de excepții de manipulare), ne vom concentra asupra şabloanelor fluxurilor de date. Perspectiva de date clasifică datele în patru categorii:

a) *vizibilitatea datelor* (exprimă vizibilitatea elementelor de date în cadrul unui proces),

b) *interacțiunea datelor* (descrie modul în care elemente de date comunică cu componentele procesului),

c) *transferul datelor* (ia în considerare modul în care datele sunt transferate de la un element al procesului la altul), respectiv

d) *rutarea datelor* (se referă la modul în care elemente de date au un impact asupra perspectivei de flux a activităților).

Concluzii capitol

Acest capitol prezintă schimbările suferite de sistemele informatice de-a lungul timpului. Dacă la începutul anilor ’70, sistemele informatice puneau mare accent pe date (de exemplu bazele de date relaționale), anii ’90 deschid cererea spre abordări orientate pe procese (de exemplu Sistemele de Management ale Fluxurilor de Lucru).

Procesele de afaceri descriu modul în care o companie îşi organizează activitățile cu scopul de a furniza produse şi servicii de valoare. Performanța organizațională poate fi îmbunătățită cu ajutorul Reproiectării Proceseselor de Afaceri. Automatizarea proceselor de afaceri face ca o organizație să fie mult mai eficientă. Aşadar, Managementul Fluxurilor de

Lucru ajută la executarea simultană a activităților prin dimunarea întregii durate de execuție a procesului de afaceri.

Sistemele de management ale Fluxurilor de Lucru propun tratarea proceselor de afaceri bazate pe cazuri pentru analiza diferite perspective: a) perspectiva fluxurilor activităților, perspectiva fluxurilor de date, respectiv perspectiva resurselor. Ultimul deceniu a fost inundat de analiza fluxului de activități. De asemenea, au fost tratate şi aspecte legate de perspectiva resurselor. Dar, o importanță minora s-a acordat perspectivei de date. În acest capitol am introdus aspect generale legate de perspectiva de date, aspecte mai detaliate urmând a fi furnizate în următorul capitol.

2. Evoluția analizei datelor

În [26] am prezentat o importantă parte a evouției studiului analizei datelor. În acest sens, am prezentat o expertiză a cercetărilor precedente efectuate în cadrul modelării fluxului de date ale proceselor de afaceri. Am ajuns la concluzia că nicio metodă, tehnică sau model analizat nu se concentrează pe mişcarea datelor de-a lungul execuției unui proces.

Figura 3 ilustrează interesul arătat față de analiza datelor de-a lungul timpului. Analiza datelor a început cu studiul Diagramelor Structurale în 1969. La puțin timp dupa aceasta, a fost definit conceptul Diagramă Entitate Relație [12]. Apoi, sistemele de tip process-aware şi-au făcut simțită prezența şi au fost dezvoltate noi abordări concentrate asupra datelor. Pentru analiza fluxului de date al unui process, cercetătorii au încercat să furnizeze vizualizări de sine stătătoare ale perspetivei acestuia sau au combinat fluxul de activități cu cel al datelor. În acest sens, au fost dezvoltate metode de validare ale datelor şi au fost identificate câteva erori care pot fi întâlnite la nivelul fluxului de date.

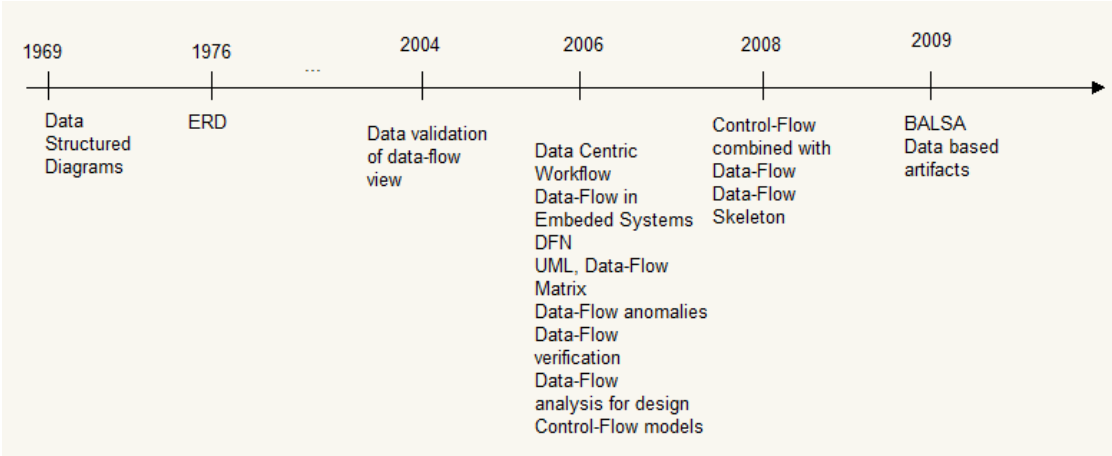

Figure 3 Evoluția analizei datelor

Modelarea datelor se referă la analiza de structuri orientate pe date (de exemplu bazele de date relaționale). Mai întâi de toate, în proiectarea bazelor de date relaționale, cel mai cunoscut şi utilizat model pentru descrierea elementelor de date şi interacțiunea acestora este diagrama Entitate Relație (ERD) [12]. Un astfel de model descrie *entitățile* , *atributele* lor şi *relațiile* dintre entități. Diagramele ER descriu un model de date static, în timp ce noi dorim o modelare dinamică a perspectivei fluxului de date a unui proces (de afaceri). Prin urmare,

este evident faptul că a genera o diagramă ER dintr-un log conținând mai multe cazuri și utilizarea acesteia în legătură cu modelele de procese este imposibil de realizat.

Apoi am făcut corelații cu privire la modul în care Diagrama de Activități poate reprezenta perspectiva de date. Dar aceasta oferă detalii legate de ordinea activităților în cadrul unui proces. Deci, folosind acest tip de diagrame în contextul fluxurilor de lucru este nefezabilă.

O tehnică ce combină Diagrama de Activități UML cu o abordare bazată pe date este descrisă în [31]. Mai întâi, Diagrama de Activități este extrasă, iar apoi elementele de date sunt împărțite în două categorii în funcție de activitatea de care aparțin: elemente de date de intrare dacă acestea sunt citite în cadrul activității curente, respectiv elemente de date de ieşire dacă acestea sunt srise în cadrul activității curente. Elementele de date ajută la crearea matricei fluxului de date. Apoi aceasta matrice este integrată în cadrul perspectivei fluxului de activități (desris de către Diagrama de Activități) şi rezultă Diagrama de Date a Procesului. Neajunsul acestei metode constă în neoferirea unei metode automate pentru furnizarea perspectivei fluxului de date. Mai mult, aceasta metodă nu oferă un model pur bazat pe date deoarece perspectiva fluxului de date este combinată cu cea a activităților. O abordare apropiată de Diagrama de Date a Procesului este reprezentată de Scheletul Fluxului de Date decorat cu Activități (DFSFA) [15]. Fluxul de lucru al procesului este derivat din scheletul fluxului de date şi apoi este decorat cu activități. În primul rând este construit un arbore al dependenței fluxului de date. Următorul pas este generarea scheletului fluxului de date după care este decorat cu activități. Principala diferență față de abordarea noastră este faptul că noi luăm în considerare logurile de evenimente produse de către sistemele informatice şi într-un mod automat extragem dependențele de date clasificându-le în elemente de intrare şi de ieşire, în timp de în [15] şi în [30], dependențele dintre date sunt extrase analizând semantica. Cu alte cuvinte, datele de intrare, respectiv de ieşire sunt ştiute de la început de către proiectantul procesului.

Una dintre cele mai cunoscute diagrame UML propusă la începutul anilor '70 este Diagrama de Flux a Datelor (DFD). Această diagramă arată procesele generate de către un sistem informatic prin combinarea proceselor (activităților) cu entitățile externe (resurse) și cu datele (obiecte de date și depozite de date). Principala problemă este că o DFD descrie doar activitățile care sunt direct legate de prelucrarea datelor, prin urmare, activitățile care nu implică nicio modificare a datelor nu vor fi incluse într-o DFD. Mai mult decât atât, niciun model de date pur nu este oferit deoarece DFD combină resursele cu activitățile și datele.

Metagrafele [9], [10] reprezintă o altă tehnică de modelare care combină elemente de date cu activitățile fluxurilor de lucru. Prin urmare, nu prezintă o abordare bazată pur pe date, dar adevăratul neajuns apare atunci când avem de-a face cu metagrafe complexe: ele sunt dificil de citit, respectiv de analizat.

În continuare vom prezenta o abordare apropiată de abordarea noastră: Proiectarea Fluxurilor de Lucru Bazate pe Produse. Mai întâi vom descrie Lista materialelor așa cum este definită în [22]. Această noțiune este folosită, în general, în ingineria de fabricație și are o structură arborescentă. În [3] definiția Listei materialelor a fost extinsă cu opțiuni și alegeri. Acesta a fost punctul de pornire, în definirea modelului de tip Product Data (PDM) [33]. Un model de tip Product Data oferă o vizualizare a unui flux de lucru concentrată pe date (vezi Figure 4). Principala problemă a acestei abordări este că nu propune nicio metodă automată

pentru extragerea PDM-ului. Deoarece compoziția manuală a activităților este o mare consumatoare de timp, chiar și pentru persoanele de specialitate, o compoziție a activităților în contextul Proiectării Fluxurilor de Lucru Bazată pe Produse este introdusă în [1]. Mai mult, algoritmul care permite generarea automată a activităților pentru un PDM dat este integrat în Platforma ProM. Având reprezentarea XML a modelului PDM, fișierul XSD poate fi importat în Platforma ProM. Definiția XML este formată din setul de elemente de date, respectiv setul de operații. Algoritmul returnează setul de activități pe baza setului de elemente de intrare, al setului de elemente de ieșire, respectiv al setului de operații. Mai mult, returnează o nouă vizualizare modelului PDM inițial, luând în considerare cele mai importante elemente de date ale acestuia. Punctul comun cu abordarea noastră este dat utilizarea Platformei ProM pentru a furnizarea vizualizarii bazate pe date, în timp ce principala diferență este că abordarea noastră propune pentru fiecare activitate din proces (o activitate dintr-un proces reprezintă un eveniment din logul de evenimente rezultat) o singură operație în PDM-ul corespunzător.

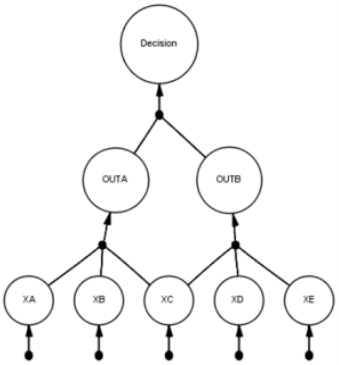

Figure 4 Structura Product Data Model

În continuare vom introduce unele metode și tehnici utilizate pentru verificarea fluxului de activități și de date. Chiar dacă nu este propus un model pur bazat pe date, au fost definite unele tehnici de verificare ale fluxului de date [28]: date redundante, date pierdute, date lipsă, date nepotrivite, date inconsistente, date rătăcite și date insuficiente.

Există mai multe metode și tehnici de detectare a erorilor fluxului de activități în proiectarea fluxului de lucru [5], [6], [29] care sunt implementate în cadrul unor instrumente (Woflan [7], [34], [36]), în timp ce verificarea fluxului de date presupune o analiză detaliată a dependențelor datelor [32]. Blocaje, robustețe, cicluri interminabile și sincronizarea sunt câțica termeni care aparțin ariei de verificare a fluxului de activități. Mai mult decât atât, o serie de anti-șabloane referitoare fluxul de date sunt introduse în [32]: lipsa datelor, date puternic redundante, date slab redundante, date puternic pierdute, data slab pierdute, date inconsistente, date niciodată distruse, date distruse de două ori și date neeliminate la timp.

De asemenea, literatura oferă un algoritm de descoperire a erorilor fluxului de date menționate mai sus [37]. Algoritmul GTforDF (Grafic de Traversare a Fluxului de Date), oferă o abordare care traversează fluxul de lucru analizat în scopul de a găsi toate cazurile și de a găsi erorile fluxului de date. Pentru fiecare activitate a unui caz sunt definite seturi de intrare și de ieșire.

Apoi am descris modelele de flux de date (grafice care descriu programele într-un mod grafic). Acestea se apropie de prima noastră abordare de extragere a modelui fluxului de date sub forma unei vizualizări grafice (modelul fluxului de date foloseşte cercuri pentru reprezentarea operanzilor în locul elementelor de date). Acesta utilizează unele operații aritmetice (un program) ca input, în scopul de a asigura vizualizarea în locul utilizării unui log de evenimente. Dar logurile de evenimente nu se referă numai la operațiile aritmetice, ele constau în activități (și datele se modifică la nivel de activitate). Orice modificare a datelor poate fi văzută ca o operație.

Concluzii capitol

Acest capitol prezintă câteva metode şi tehnici de analiză a datelor sau modele axate pe vizualizarea fluxului de date a unui proces. Am început cu tipuri de modelare de bază a datelor (de exemplu ERD) deoarece ele stau la baza proiectării bazelor de date relaționale. De asemenea, am amintit câteva abordări care pun accent pe perspectiva fluxului de date (de exemplu Diagrama de Activități UML), în scopul de a face trecerea la abordări care combină perspectiva fluxului de date cu perspectiva fluxului de activități (de exemplu Diagrama de Date a Procesului).

Apoi am descris modelul Product Data așa cum a fost definit în [33] şi modul în care se potriveşte abordării noastre. O abordare care utilizează modelele de tip PDM este descrisă în [1]: etapele de prelucrare a datelor sunt grupate în activități. Fişierul XSD care urmează a fi importat în ProM constă într-un set de activități şi un set de elemente de date [33]. Dezavantajul acestei abordări este că ea consideră fişierul PDM XML deja creat şi pe baza aceastuia este construită agregarea. Scopul nostru este de a crea în mod automat modelul PDM (în primul rând urmărim crearea fişierului XML în format XES, iar mai apoi vizualizarea acestuia).

Modelul fluxului de date a fost definit pentru a oferi o vizualizare grafică a unui program. Aceasta vizualizare este similară cu vizualizare modelelor de tip PDM (foloseşte cercuri pentru elementele reprezentate în cadrul programului, de exemplu, pentru operanzi). Abordarea noastră foloseşte un log de evenimente ca input. Dar logurile de evenimente nu se referă numai la operații aritmetice, ele constau în activități (și datele se modifică la nivel de activitate). Orice modificare a datelor poate fi văzută ca o operație.

Câteva anti-şabloane ale fluxului de date sunt formalizate în [32]. Ele descriu erori care pot apărea în fluxul de date oferă metode pentru descoperirea acestora (de exemplu, şablonul lipsei datelor).

Am ajuns la concluzia că niciuna dintre metodele, tehnicile sau modelele analizate nu subliniază mișcarea (sau modificarea) datelor în cadrul execuției unui proces. Mai mult decât atât, niciuna dintre abordările prezentate nu foloseşte ca punct de plecare un log de evenimente pentru a analiza perspectiva de date.

3. Process mining

Acest capitol face introducerea spre domeniul process mining (vezi Figura 5), prin prezentarea principalelor noțiuni ale acestuia. Process mining-ul combină tehnici de modelare şi analiză a proceselor cu tehnici de data mining și învățare automată [4]. Acesta constă într-un set de tehnici și metode care oferă informații despre procesul analizat.

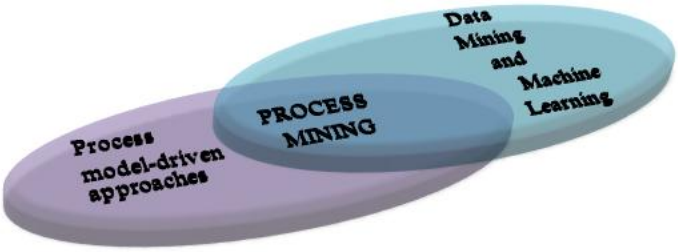

Figure 5 Domeniul Process mining

Tehnicile de process mining extrag cunoştințe şi modele din logurile de evenimente. Pe lângă acestea, folosind tehnici din sfera process mining-ului, procesul de afaceri poate fi monitorizat sau îmbunătățit. Fiecare eveniment se referă la o activitate a procesului. Acesta stochează informații despre numele activității, resursa care a finalizat activitatea, momentul în care a avut loc sau orice alte informații referitoare la activitatea respectivă. Totalitatea evenimentelor execuției unui proces formează un caz (o instanță de proces). Un caz se referă la o singură execuție a procesului. Mai multe cazuri formează logul de evenimente (vezi Figura 6). Logul de evenimente prezentat mai jos are în componența sa *t* cazuri. Fiecare caz este compus dintr-un anumit număr de evenimente (aceasta poate diferi de la un caz la altul; de exemplu cazul 1 are *m* are evenimente, în timp ce cazul *t* are *r* evenimente). Numărul de atribute ale fiecărui eveniment poate diferi de la un caz la altul, de exemplu, evenimentul *r* din cazul *t* are *s* atribute.

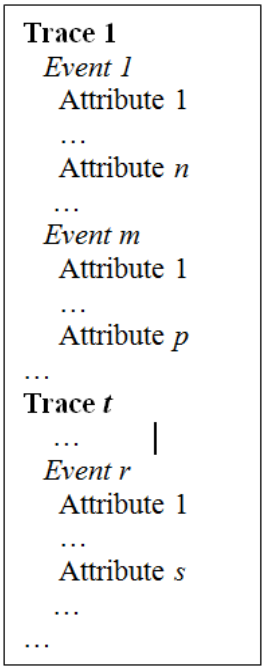

Figure 6 Componentele unui log de evenimente

Domeniul process mining reprezintă un domeniu larg şi implică mai multe abordări: descoperirea proceselor, verificarea conformității proceselor şi îmbunătățirea acestora. Descoperirea proceselor se referă la algoritmi de descoperire care extrag cunoaştințe prin intermediul modelului minat. Aceasta implică existența unui log de evenimente pe baza căruia este construit un model. Modelele minate pot avea diferite abordări:

a) *perspectiva fluxului de activități*: accentul este pus pe ordinea activităților (de exemplu Alfa Miner [4], [8], [20], Fuzzy Miner [17], Heuristics Miner [38], Genetic Miner [21]);

b) *perspectiva organizatorică*: se concentrează asupra relațiilor dintre resurse (de exemplu Social network miner [2]).

Verificarea conformității proceselor [27] are ca scop descoperirea de abateri între comportamentul dorit al unui proces şi comportamentul real al aceluiaşi proces. Există două moduri pentru a face acest lucru:

a) modelul procesului inițial este comparat cu modelul minat sau

b) analiza modului în care modelul inițial sau modelul minat este capabil de a reproduce logul de evenimente inițial.

Pentru aplicarea tehnicilor de process mining asupra logurilor de evenimente o serie de instrumente sunt necesare. Există diferite instrumente pentru extragerea cunoştințelor din logurile generate de execuția proceselor. Problema este că acestea folosesc diferite formate de intrare şi modelele rezultate sunt reprezentate în mai multe moduri. Pentru a rezolva acest neajuns, un grup de cercetători de la Universitatea Tehnică din Eindhoven a dezvoltat o platformă generică numit ProM. Platforma ProM acceptă mai multe tehnici de process mining sub forma de plug-in-uri. Mai mult decât atât, ProM este capabil de a importa loguri de evenimente în conformitate cu formatele MXML [13], [14], [16] sau XES [18].

Fiecare sistem de informatic de tip process-aware, are propria structură de date. Din această cauză logurile de evenimente produse de sisteme de informatice de tip process-aware diferite au formate diferite. Acesta este motivul pentru care este nevoie de un mod standardizat pentru vizualizarea aceste loguri de evenimente. Primul format standard propus a fost MXML (Mining Extensible Markup Language [13, 14, 16]). Acesta reprezintă un format bazat pe XML pentru stocarea logurilor de evenimente şi are o notație standard pentru stocarea de marcajelor de timp, resurselor şi tipurilor de tranzacții. Se pot adăuga elemente de date pentru evenimente şi cazuri. Succesorul MXML este Event Stream eXtensible (XES). Scopul standardului XES este de a oferi portabilitate pentru data mining, text mining, şi analiza statistică [18]. Plug-in-urile dezvoltate în cadrul Platformei ProM 6 sunt grupate în pachete şi orice pachet poate fi instalat şi actualizat în mod independent cadrul platformei.

În comparație cu formatul MXML, extensia XES consideră toate atributele pe acelaşi nivel. Fiecare atribut poate avea un alt tip (de exemplu: string, data, întreg, float sau boolean). Lucrarea [18] descrie aceste tipuri de atribute. O serie de extensii au fost definite în scopul de a ataşa semantică datelor. Fiecare extensie are un anumit număr de atribute. Acestea pot fi definite la nivelul logului, cazului sau evenimentului. Mai mult decât atât, un atribut poate conține, la rândul său, alte atribute (meta-atribut). Aici vorbim despre atributele imbricate. Fiecare extensie are trei atribute obligatorii : "*nume*", "*prefix*" şi "*uri*". Atributul

"*nume*" se referă la numele extensiei. Atributul "*prefix* " face legătura între extensiile declarate la nivelul logului şi extensia folosită la nivel de caz, respectiv la nivel de eveniment. Prefixul "*uri*" deține calea pentru definiția extensiei. Extensiile standard actuale [18], [35] sunt: extensia concept, extensia ciclului de viață, extenia organizațională, extentia timpului și extensia semantică.

Concluzii capitol

Acest capitol introduce noțiunea de process mining, o abordare care combină elemente din modelarea proceselor, data mining și învățare automată. Punctul de pornire al process mining-ului este logul de evenimente și scopul acestuia este de a extrage cunoștințe din logurile de evenimente.

Logurile de evenimente oferite de sistemele informatice de tip process-aware sau Sistemele de Management a Fluxului de Lucru folosesc diferite formate de intrare. Acesta este motivul pentru care a fost creată Platforma ProM: pentru a oferi un instrument unic care include o serie de algoritmi de process mining.

De obicei, plug-in-urile implementate în cadrul Platformei ProM reprezintă algoritmi de mining care analizează un log de evenimente (de exemplu, oferă o vizualizare a modelului de proces descris în logul de evenimente).

În scopul de a asigura un mediu unificat pentru process mining două standarde au fost definite: MXML și XES. Ne-am concentrat pe standardul XES și extensiile sale (de exemplu, extensia concept), în scopul de a asigura vizualizarea fluxului de date. Astfel, logul de evenimente folosit ca punct de plecare pentru perspectiva fluxului de date utilizează formatul XES.

Încheiem acest capitol prin analiza unor plug-in-uri puse în aplicare în cadrul Platformei ProM. Niciunul dintre aceştia nu oferă o vizualizare a mişcării datelor în cadrul unui proces, cele mai multe dintre ele concentrându-se pe perspectiva activităților (de exemplu Alpha Miner, Fuzzy Miner). Chiar dacă există câțiva algoritmi care iau în considerare unele date prezente în cadrul procesului, aceştia nu oferă o perspectivă a fluxului de date.

4. Algoritmi de mining

Acest capitol prezintă contribuția principală a tezei de doctorat prin descrierea algoritmilor bazați pe date. Figura 7 evidențiază etapele urmate pentru crearea modelului fluxului de date. Am creat două instrumente de conversie deoarece logurile de evenimente generate de sistemele informatice nu sunt conforme cu formatul Fluxului de Date: *Convertorul Fluxului de Date* și *Convertorul I/O*. În primul rând am definit formatul XML pentru logurile de evenimente ale Fluxului de Date (formatul XES). Prin urmare, am definit o extensie care caracterizează logurile de evenimente ale Fluxului de Date (extensia *InputOutput*). Având formatul XES dorit putem analiza logurile de evenimente generate de sisteme informatice de tip decision-aware și cele generate de sistemul YAWL prin aplicarea algoritmilor de mining bazați pe date. Logurile de evenimente generate de sistemul informatic de tip decision-aware nu respectă formatul XES, dar ele conțin elementele de date

necesare. Acestea sunt în conformitate cu formatul vechi propus pentru logurile de evenimente, şi anume formatul MXML.

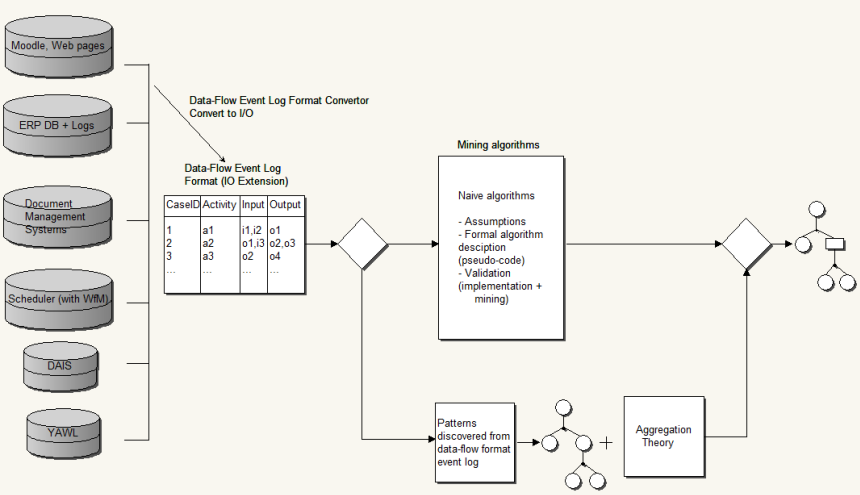

Figure 7 Abordare generală

În scopul obţineruu perspectivei fluxului a datelor unui process propunem două abordări:

a) prima abordare oferă un PDM pentru fiecare caz al logului de evenimente, respectiv

b) a doua abordare oferă un PDM pentru întregul log de evenimente.

În primul caz, este posibilă o agregare a tuturor cazurilor (sau a unui anumit număr de cazuri dorite de către utilizator). Mai multe detalii pot fi găsite în lucrarea [24]. În primul rând vom începe cu dezvoltarea unei aplicaţii care oferă perspectiva de flux a datelor din proces având ca punct de pornire un singur trace al unui log de evenimente. Apoi vom descrie algoritmii naivi integraţi în cadrul Platformei ProM.

Prima abordare utilizează loguri de evenimente generate de către un sistem informatic de tip decision-aware. Pentru fiecare execuţie a procesului descris un acesta un PDM este generat. Această abordare are ca input un trace dintr-un fişier XML care conţine toate operaţiile înregistrate de către utilizator într-un software de tip web-based. Software-ul reprezintă o cerere pentru un credit. Acesta poate fi utilizat de către managerul unei companii, în scopul de a lua o decizie cu privire la un credit. Software-ul oferă informaţii despre activitatea companiei pentru un întreg an. Decidentul poate lua o decizie completă de finanţare în ceea ce priveşte împrumutul sau o decizie referitoare la împrumut: perioada de creditare, tipul de credit sau altele.

Logurile de evenimente generate de interacţiunea utilizatorului cu software-ul sunt stocate într-o bază de date formată din cinci tabele şi sunt în conformitate cu formatul MXML descris în capitolul anterior. Aplicând algoritmul de mining asupra logurilor de evenimente rezultate, un PDM este generat în mod automat. Acest capitol oferă aspecte cu privire la pseudo-codul generării fişierului PDM-XML. Acest fişier trebuie să fie importat în Platforma ProM (5.2 sau mai nouă), pentru a obţine vizualizarea grafică. Aşa cum am argumentat mai devreme, pentru fiecare caz un PDM este construit.

Pentru validarea primei noastre abordări am fost utilizate mai multe exemple de funcţionare şi patru cazuri de utilizare. Structura fişierului PDM-XML este creată pe baza

următoarelor elemente: BD (date de bază), ID (date introduse), DD (date derivate), şi O (output). Următorul pas este de a defini operaţiile bazate pe BD, ID-ul şi DD. Odată ce aceste elemente sunt extrase, fişierul PDM - XML poate fi generat şi importat în Platforma ProM pentru a obţine vizualizarea bazată pe date.

A doua parte a celui de-al patrulea capitol al tezei oferă detalii despre cei trei algoritmi naivi implementaţi pentru generarea perspectivei fluxului de date: Algoritmul Naiv A, Algoritmul Naiv B şi Algorithmul Naiv C. Fiecare algoritm produce o vizualizare grafică diferit. Algoritmul Naiv A se concentrează pe un set de activităţi existente, aşa cum apar în logul de evenimente. Cel de-al doilea algoritm compară activitatea curentă din log cu activităţile anterioare, luând în considerare elementele de date de intrare, respectiv elementele de date de ieşire. În cele din urmă, Algoritmul Naiv C include probabilitatea ca o operaţiune să genereze mai multe elemente de date de ieşire.

Aşa cum am argumentat mai devreme, algoritmii de mining bazaţi pe date se aplică unor loguri de evenimente care trebuie să respecte formatul Fluxului de Date. Prin urmare, am definit o extensie care caracterizează logurile de evenimente de tip Flux de Date (extensia *InputOutput*). Această extensie defineşte elementele imbricate, în scopul de a asocia elemente de date de intrare, respectiv elemente de date de ieşire pentru fiecare eveniment. Mai mult decât atât, această definiţie asociază două elemente de date pentru fiecare eveniment (*de intrare* şi *de ieşire*). Pentru fiecare element de *intrare* sunt definite două elementele date: *nume* şi *valoare*. În acelaşi mod, un anumit element de *ieşire* conţine două elemente de date. Aşa cum numele sugerează elementul *nume* se referă la numele elementelor de date de intrare/ieşire, în timp ce elementul *valoare* stochează valoarea elementului de date de intrare/ieşire. Având în vedere faptul că un eveniment poate conţine mai mult de un element de tip intrare sau ieşire am ales să delimităm elementele de date prin folosirea diezului ("#"). Această extensie defineşte numai elementele de date necesare pentru fiecare activitate, în timp ce numele operaţiei poate fi găsit în extensia concept.

În continuare vom prezenta algoritmii de mining bazaţi pe date. Pentru fiecare algoritm am oferit o definiţie formală, pseudo-codul aferent şi o serie de ipoteze specifice (de exemplu: mai multe elemente de date de intrare, un singur element de date de ieşire, mai multe elemente de date de ieşire, frecvenţa, etc).

IDCaz	Activitate	Elemente de date de intrare	Elemente de date de ieşire
1	registration	fName, lName, title, street, City, pCode, country, phone, eMail	newClient
1	roomInformation	newClient, roomNo, rateType, noNights, noAdults, noChildren and notes	roomInfo
2	registration	fName, lName, title, street, City, pCode, country, phone, eMail	newClient
2	roomInformation	newClient, roomNo, rateType, noNights, noAdults, noChildren	roomInfo

Table 1 Cazuri utilizate pentru Algoritmii Naivi A şi B

Algoritmul naiv A identifică operaţiile în mod unic, pe baza denumirilor lor. În cazul în care există o activitate care are acelaşi nume, dar diferite elemente de date de intrare şi/sau diferite elemente de date de ieşire, operaţia care va fi reprezentată în cadrul model va fi prima găsită în logul de evenimente. Pentru Algoritmul Naiv B, o operaţie este definită cu ajutorul elementelor componente: setul de elemente de date de intrare, respective setul de

elemente de date de ieşire. În primul rând căutam elemente de date de intrare şi de ieşire în logul de evenimente şi apoi atribuim setul de elemente de intrare/ieşire operaţiilor. Prin urmare, operaţiile cu diferite elemente de date de intrare (ieşire) vor fi afişate în vizualizarea PDM-ului. Ultimul algoritm (Algoritmul Naiv C) se concentrează pe mai multe elemente de date de ieşire (funcţionalităţile sale sunt aceleaşi cu funcţionalităţile de Algoritmulului Naiv B, diferenţa fiind că primul algoritm amintit acceptă elemente de date de ieşire multiple).

În continuare vom descrie diferenţele dintre primii doi algoritmi. Logul utilizat este prezentat în tabelul 1, în timp ce vizualizarea de date în Figura 8.

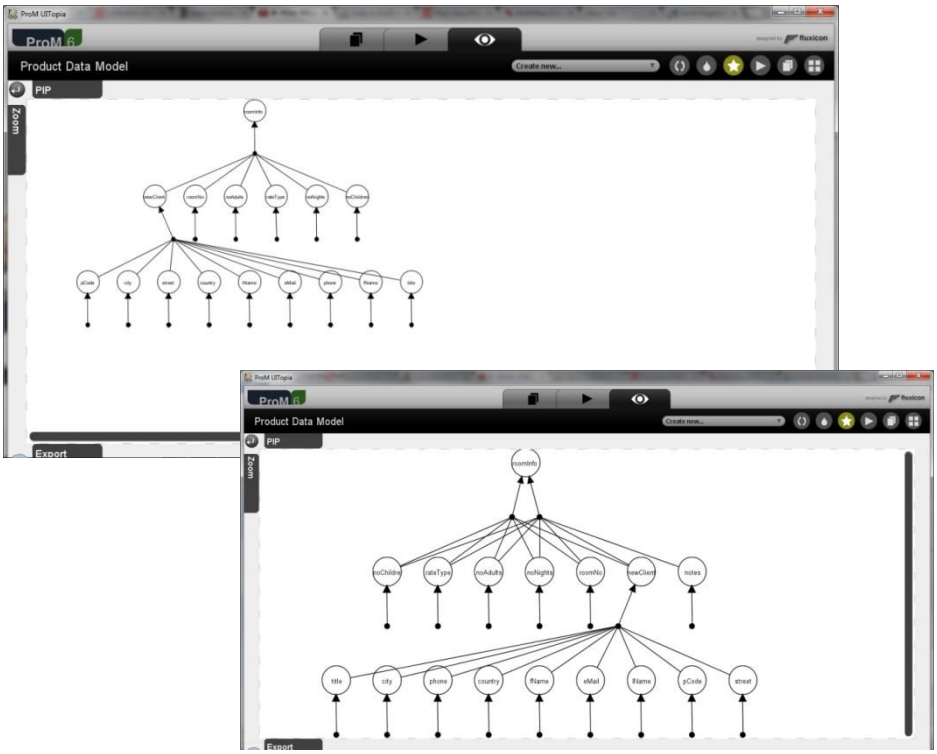

Figure 8 Algoritmul Naiv A versus Algoritmul Naiv B aplicat lgului din tabelul 1

În continuare ne vom concentra pe convertoare dezvoltate: *Convertorul Fluxului de Date (DFC)* şi *Convertorul spre I/O log*. Primul se aplică logurilor de evenimente produse de sistemele de tip ERP şi nu este complet automatizat în timp cel de-al doilea se referă la logurile de evenimente cu evenimente *start* şi *complete* (de exemplu logurile de evenimente produse de sistem YAWL).

Primul pas al *DFC* este de a determina procesul care urmează a fi analizat. În această fază, specialiştii de afaceri (de exemplu analiştii de afaceri , managerii , şefii de departamente etc.) derivă etapele procesului pe baza procedurilor interne şi pe baza experienţei lor. Apoi, am pus dispositive de urmărire (tracker-e) pe care tabelele care furnizează datele necesare pentru execuţia procesului. O data ce avem logul de evenimente generat de sistem putem aplica algoritmul Fluxului de Date. În primul rând, avem de definit clusterele pentru procesul nostru. Acestea sunt extrase din activităţile procesului (de exemplu pentru procesul *Comanda-Plată* putem avea următoarele clustere: creează comandă, generează facturi şi încasare factură). Practic fiecare activitate reprezintă un cluster. Odată ce avem clusterele create, putem analiza fiecare dintre acestea. Pentru fiecare grup toate datele sunt colectate

(inclusiv numele şi valorile acestora). Prin urmare, acestea sunt elementele de date de intrare conform extensiei *InputOutput*. În ceea ce priveşte elementul de date de ieşire vom crea un element de date artificial care are ca nume iniţialele clusterului de care aparţine. De asemenea, se va păstra valoarea aferentă.

Cel de-al doilea convertor (*Convert to I/O log*), transformă evenimentele de tip start şi complete în evenimente conforme cu formatul Fluxului de Date. Plug-in-ul de conversie este integrat în cadrul Platformei ProM ("*Convert to I /O log*" plug-in). Foloseşte ca şi punct de pornire un log de evenimente care conţine evenimente de tip start şi complete (în format XES) şi returnează un log de evenimente corespunzătoare în formatul Fluxului de date (cu elementele de date de intrare şi ieşire pentru fiecare eveniment). Logurile de evenimente de tip *start* şi *complete* fac separarea elementelor de date mai uşoară. În plus, elementele de date artificiale nu sunt mai necesare (ca în abordarea de conversie anterioară - DFC).

Concluzii capitol

Acest capitol prezintă contribuţia principală a tezei de doctorat. În acest context sunt definite două abordări: a) prima abordare oferă un PDM pentru fiecare caz din logul de evenimente şi b) a doua abordare oferă un PDM pentru întreg log de evenimente. Cea de-a doua abordare suportă ca intrare un log de evenimente în formatul de intrare/ieşire. Pe această direcţie, formatul de intrare/ieşire a fost definit folosind standardul XES. Acesta este formatul standard suportat de Platforma ProM (pe lângă format MXML). Platformei ProM îi lipsesc plug-in-uri care să ofere o vizualizare a fluxului de date, cele mai multe de plug-in-uri implementate punând accent pe perspectiva fluxului de activităţi (de exemplu, α-Miner, Fuzzy Miner, Heuristics Miner etc.). Pentru fiecare algoritm care oferă perspectiva de flux a datelor a fost implementat un plug-in în cadrul Platformei ProM şi fiecare plug-in oferă o vizualizare diferită a procesului.

Pentru a se conforma cu format Fluxului de Date (şi extensia *IO*), două convertoare sunt dezvoltate: *Convertorul Fluxului de Date* şi *Convertorul spre I/O log*. Primul se aplică logurilor de evenimente produse de sistemele ERP şi nu este complet automatizat în timp cel de-al doilea se referă la logurile cu evenimente de tip *start* şi *complete* (de exemplu logurile de evenimente produse de sistemul YAWL).

5. Şabloane de date

În scopul validării şabloanelor de date ale modelelor de tip PDM am folosit logurile de evenimente din [4]. Aceste loguri au fost create în scopul de a se concentra asupra perspectivei fluxului de activităţi. Prin urmare, în logul de evenimente au fost descrise numai datele referitoare la numele activităţii, resursa care a executat activitatea, timpul când activitatea a fost efectuată. De asemenea, datele sunt folosite şi pentru condiţiile de rutare. Am îmbunătăţit aceste loguri de evenimente prin completarea date suplimentare pentru fiecare eveniment, în scopul de a evidenţia perspectiva de date. Prin urmare, am folosit extensia *InputOutput* definită. Fiecare eveniment este caracterizat prin elemente de date de intrare şi de ieşire. Chiar dacă am introdus noi elemente de date pentru fiecare eveniment din log fluxul de activităţi a rămas acelaşi.

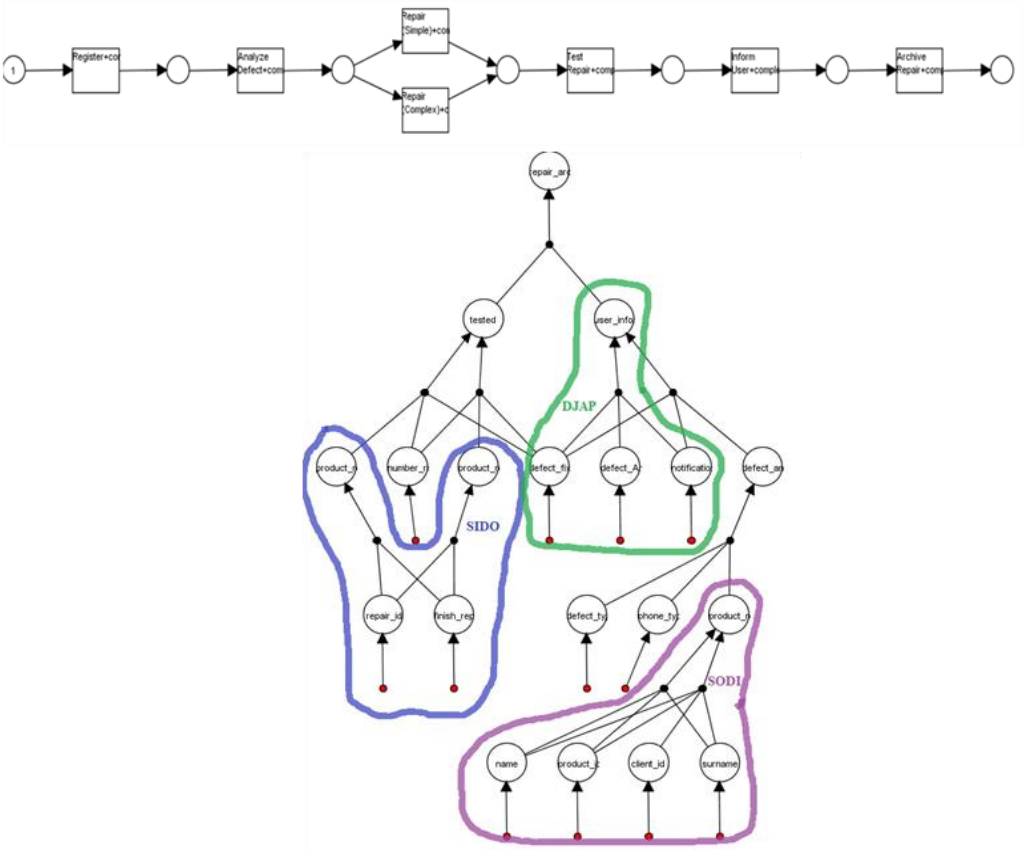

Figure 9 Fluxul activităților versus Fluxul datelor (şabloane de date)

Figura de mai sus arată şabloanele de date care pot fi identificate în cadrul unei vizualizări bazată pe date. Este evident faptul că şabloanele de date nu pot fi identificate în cadrul perspectivei fluxului de activități. În acest caz, şablonul XOR-Split, din perspectiva fluxlui de activități este reprezentat de şablonul SIDO din perspectiva fluxului de date.

Am introdus cinci şabloane de date identificate la nivelul evenimentelor (de exemplu Îmbinarea Datelor din Atribute- DJAP, acelaşi element de date de ieşire, elemente de date de intrare diferite - SODI, actualizări de date - DU), dar aceste şabloane de date pot fi, de asemenea, găsite la nivel de caz. Şabloanele SODI şi SIDO oferă alternativa pentru şablonul Exclusive Choice (XOR-Split) din fluxul activităților.

Concluzii capitol

În acest capitol este descrisă o serie de şabloane de date (de exemplu a) şabloane de date de bază (BP): şablonul Îmbinării Datelor din Atribute (DJAP), b) şabloane complexe de date: şablonul aceeaşi intrare, ieşire diferită (SIDO), aceeaşi ieşire, intrare diferită (SODI), şi c) şabloane ale valorilor de date: actualizări de date (DU), valoare de date condiționată (CDV)). Mai mult decât atât, este descrisă o formalizare a acestor şabloane de date.

De asemenea, şabloanele de date oferă o reprezentare pe baza de date a şabloanelor fluxului de activități, cum ar fi şablonul Alegerii Exclusive (XOR-Split). Propunem două reprezentări bazate pe date pentru Alegerea Exclusivă prin şabloanele *aceeaşi ieşire, intrare diferită (SODI)* şi *aceeaşi intrare, ieşire diferită (SIDO)*. Cel mai potrivite pentru şablonul XOR-Split este primul şablon (SODI) pentru că atunci când avem două activități ale căror elemente

de date de ieşire sunt la fel, de asemenea, operaţiile (activităţile) sunt aceleaşi, în timp ce în cazul în care elementele de date de ieşire sunt diferite, nici activităţile nu sunt aceleaşi.

6. Studii de caz

Al şaselea capitol validează algoritmii utilizând două studii de caz (vezi Figura 10). Pentru fiecare studiu de caz au fost folosite diferite instrumente de conversie în scopul de a dovedi generalitatea acestora. Primul studiu de caz utilizează convertorii XESame si DFC pentru a genera logurile necesare deoarece sursa de date este dată sub formma unor fişiere Excel (.xlsx); în timp ce pentru al doilea studiu de caz un nou instrument de conversie a fost implementat pentru crearea logurilor de evenimente utilizând evenimente de tip *start* şi *complete*.

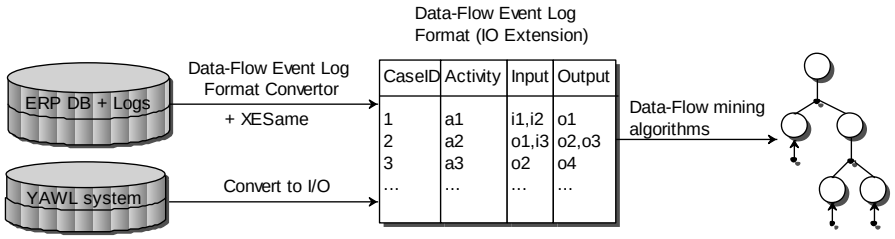

Figure 10 Abordările studiilor de caz

Aceste instrumente de conversie transformă logurile în formatul dorit (formatul Fluxului de Date). Prin urmare, putem aplica algoritmii naivi bazaţi pe date. Pentru fiecare studiu de caz am prezentat vizualizarea furnizată de un plug-in care evidenţiază fluxul activităţilor unui proces (de exemplu Alpha Miner). Apoi am aplicat algoritmii naivi şi am descris avantajele furnizate de perspectiva fluxului de date.

Logurile de evenimente utilizate pentru primul studiu de caz sunt extrase dintr-un export de date din Navision, un sistem de tip ERP utilizat de mai multe companii din România. Datele pentru acest studiu de caz au fost furnizate de Farmec[1].

Acest capitol prezintă o secţiune care descrie detaliat modul în care logurile furnizate de Navision sunt convertite în loguri cu formatul Fluxului de Date. Rezultatul astfel obţinut poate fi importat în Platforma ProM şi pot fi realizate o serie de analize. Dupa importul logului rezultat în ProM am observat că logul de evenimente are 251 de cazuri si cuprinde 523 de evenimente. În prima fază au fost aplicaţi algoritmi care evidenţiază perspectiva fluxului de activităţi. Aceste modele nu arată care date sunt consumate sau produse de o anumită activitate, chiar dacă aceste date sunt stocate la nivel de activităţii.

[1] www.farmec.ro

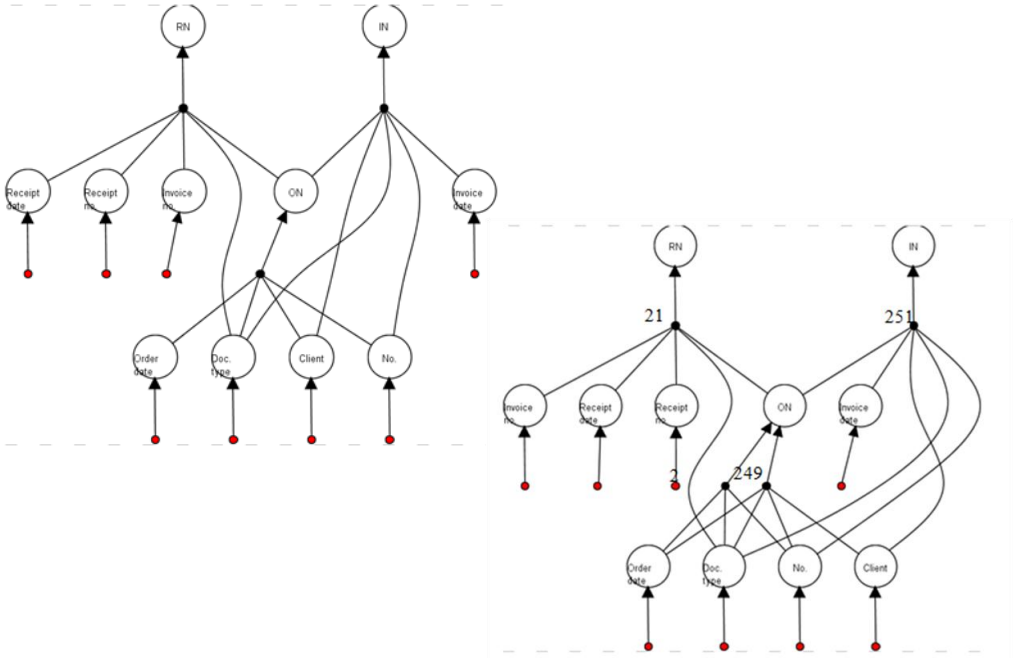

Figure 11 Naive Algorithm A versus Naive Algorithm B applied on Navision event log

Algoritmiul Naiv A se concentrează asupra operațiilor (activităților), în timp ce Algoritmul Naiv B (vezi Figura 11) se concentrează asupra elementelor de date de intrare, respectiv de ieşire a operațiilor. Chiar dacă în logul de evenimente există comenzi în cadrul cărora nu este specificat numele clientul căruia îi respectiva comandă, aceasta operație nu apare în model. Acest neajuns este rezolvat de algoritmul Naiv B deoarece acesta se concentrează pe elementele de date consumate sau produse de o activitate. Al doilea algoritm demonstrează faptul ca există două comenzi pentru care nu este cunoscut cumpărătorul. Astfel, 251 de comenzi conțin toate informațiile necesare. În plus, modelul arată că pentru fiecare comandă a fost eliberată câte o factură şi 21 dintre acestea au fost deja încasate.

Al doilea studiu de caz utilizează loguri de tip YAWL. În primul rând, am creat o specificație YAWL ilustrând procesul aprobării deplasării într-o mobilitate internațională sau națională. Apoi am fost generate logurile prin simularea procesului. Pentru o mai bună înțelegere am folosit două cazuri generate după executarea fluxului de lucru. Pe primul caz de executare a fluxului de lucru toate condițiile îndeplinite de către utilizatorii (vezi activitățile roşii din Figura 12). Pe de altă parte, pentru cea de-a doua executare, deplasarea nu este prevăzută în planul inițial (vezi activitățile verzi). Astfel, o modificare a planului inițial este necesară şi trebuie aprobată. Restul fluxului de lucru are aceeiaşi serie de pasi executați ca şi în cadrul primei execuții.

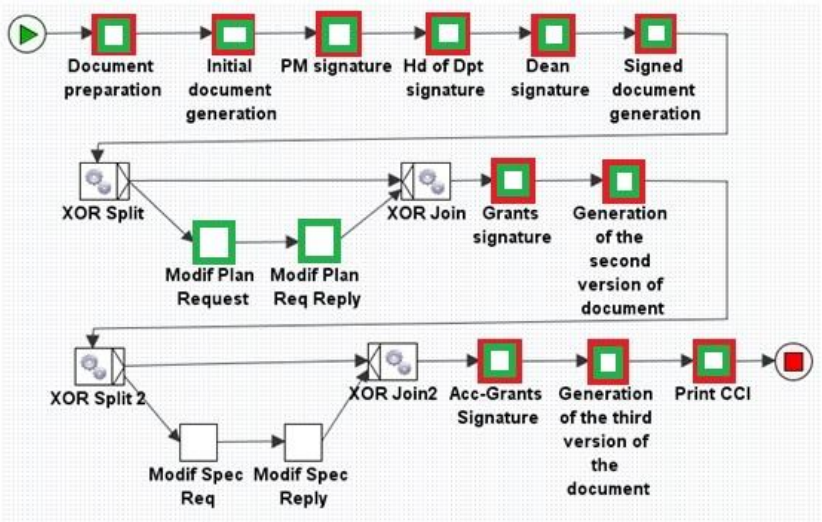

Figure 12 Fluxul activităţilor pentru procesul de deplasare într-o mobilitate internaţională sau naţională

Noi am utilizat plug-in-ul "*Convert to I/O*" asupra logurilor generate după executarea fluxului de lucru aferent specificației YAWL. Am mapat fiecare eveniment din logul de evenimente la o operație din PDM. Am setat atributele evenimentelor de tip *start* seturilor de elemente de date de intrare, în timp ce atributele de la evenimentele de tip *complete* au fost setate seturilor de elemente de date de ieşire.

Convertorul I/O nu are in vedere şabloanele XOR or (OR) din perspectiva fluxului de activități. Totusi, şablonul XOR poate fi identificat cu usurință în vizualizarea PDM-ului (vezi Figura 13).

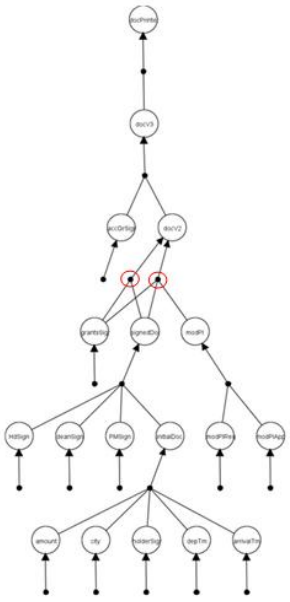

Figure 13 Şablonul XOR în modelul de tip PDM

Concluzii capitol

Două studii de caz au fost introduse în scopul validării algoritmilor propuşi. Pentru primul studiu de caz am folosit logurile de evenimente produse de un sistem de tip ERP

(Navision), în timp ce pentru cel de-al doilea am folosit loguri de evenimente generate de sistemul YAWL. Pentru a transforma logurile de evenimente din Navision în formatul dorit am folosit *XESame 1.3* şi *DFC*, în timp ce pentru logurile de evenimente generate de sistemul YAWL, a fost necesară dezvoltarea unui convertor. Vizualizarea fluxului de date a unui proces este posibilă dacă informațiile despre fiecare operație din logul de evenimente pot fi extrase (nume operației, elemente de date de intrare și elemente de date de ieșire). Ținând cont de logurile generate de YAWL, extragerea modelelor de tip PDM a fost posibilă datorită evenimentelor de tip *start* şi *complete*.

Ambele studii de caz arată informațiile suplimentare aduse de perspectiva de flux a datelor, deoarece acesta oferă mai multe informații despre process față de perspectiva fluxului de activități. Mai mult decât atât, modelele de tip PDM sunt capabile să reprezinte şabloane precum XOR și AND specifice fluxului de activități.

Concluzii şi direcții viitoare

Ultimul capitol conclude teza și prezintă direcțiile viitoare de cercetare. În general, metodele și tehnicile de process mining oferă o perspectivă a fluxului activităților unui proces, ignorând perspectiva fluxului de date. Am observat faptul că, recent, există un interes sporit demonstrat pentru perspectiva de date (de exemplu, mulți angajați de la companii reale pun întreabări de genul "dacă eu sunt într-o anumită stare a unui proces, care sunt datele cunoscute până acum și care sunt datele necesare în scopul de a finaliza procesul?"). Acest lucru ne-a motivat să urmăm o abordare de mining asupra logurilor de evenimente cu scopul de a extrage un model al fluxului de date. Am introdus contextul problemei în primele capitole ale tezei, apoi am prezentat abordarea noastră și, în cele din urmă, ne-am validat propunerile, folosind diferite loguri de evenimente.

Principalele contribuții ale tezei sunt:

• Definirea unui format standard pentru logurile de evenimente cu date;

• Dezvoltarea și implementarea unor algoritmi de mining care extrag modele care se concentrează asupra perspectivei fluxului de date;

• Implementarea a două instrumente de conversie (*DFC* și *Convert to I/O*);

• Folosirea diferitelor surse de date, în scopul validării abordarea noastre;

• Furnizarea unei alternative pentru şablonul XOR-Split în contextul perspectivei fluxului de date

Algoritmii bazați pe date se concentrează pe numele elementelor de date dintr-un proces și considerăm că fiecare element de date are aceeași valoare în timpul execuției unui proces. Dar valoarea fiecărui element de date este de asemenea importantă. Astfel, dorim să includem în cercetarea noastră analiza valorilor elementelor de date. Până în prezent, pentru un proces de execuție (caz) un element de date are aceeași valoare, dar în realitate, valoarea sa poate fi modificată în timpul execuției unui proces sau poate fi ștearsă (în acest caz, elementul de date nu va apărea în modelul rezultat). O schimbare de valoare în timpul unui proces de executare poate cauza cicluri, care nu sunt permise în conformitate cu definiția standard a modelelor de tip PDM. Ca activitate viitoare ne propunem să includem valorile datelor, în scopul de a asigura o reprezentare mai precisă axată pe date.

Concluzionăm că ne-am atins obiectivele propose furnizâmd metode automate şi semi-automate pentru obținerea perspective fluxului de date.

Bibliografie selectivă

[1] van der Aa H., Reijers H.A., and Vanderfeesten I., Composing Workflow Activities on the Basis of Data-flow Structures. In: Proceedings of the 11th International Conference on Business Process Management (BPM 2013), Lecture Notes in Computer Science, vol. 8094, pp. 275-282, Springer Verlag, Berlin, 2013

[2] van der Aalst W.M.P., "On the Automatic Generation of Workflow Processes based on Product Structures", Computers in Industry, 39:97-111, 1999

[3] van der Aalst W.M.P., "Woflan: a Petrinet- based workflow analyzer", Syst. Anal. Model. Simul., 35, 3, 345-357, 1999

[4] van der Aalst W.M. P., "Workflow verification: Finding control-flow errors using petri-net based techniques", In W.M. P. van der Aalst, J. Desel, and A. Oberweis, editors, Business Process Management: Models, Techniques, and Empirical Studies, pages 161-183. Springer-Verlag, Berlin, 2000

[5] van der Aalst W.M.P. and van Hee K.M., "Workfow Management: Models, Methods", and Systems. MIT press, Cambridge, MA, 2004

[6] van der Aalst W.M.P., Weijters A.J.M.M., and Maruster L., "Workow Mining: Discovering Process Models from Event Logs", IEEE Transactions on Knowledge and Data Engineering, 16(9):1128-1142, 2004

[7] van der Aalst W.M. P., Reijers H. A. , and Song M., Discovering social networks fromevent logs. Computer Supported CooperativeWork, 14(6):549-593, 2005

[8] van der Aalst W.M.P., Process Mining: Discovery, Conformance and Enhancement of Business Processes, Springer Verlag, 2011

[9] Basu A., Blanning R.W., "Metagraphs and Their Applications", Integrated Series in Information Systems, Senes, Springer, 2007

[10] Basu A., Blanning R.W., "Metagraphs in workflow support systems", Decision Support Systems 25, pages 199- 208, 1999

[11] BPM and Workflow Handbook, Future Strategies Inc., Layna Fischer (ed.), 2007,

[12] Chen P.P.-S. , "The entity-relationship model toward a unified view of data", ACM Transaction Database System 1, 1 pp. 9-36, 1976

[13] van Dongen B. F. and van der Aalst W. M. P., "A meta model for process mining data", In Proceedings of the CAiSE, 2005, vol. 5, pages 309-320

[14] van Dongen B., Alves de Medeiros A.K., Verbeek H.M.W., Weijters A.J.M.M., and van der Aalst W.M.P., "The ProM framework: A New Era in Process Mining Tool Support", In G. Ciardo and P. Darondeau, editors, Application and Theory of Petri Nets 2005, volume 3536 of Lecture Notes in Computer Science, pages 444-454. Springer-Verlag, Berlin, 2005

[15] Du N., Liang Y., Zhao L. , "Data-flow skeleton filled with activities driven workflow design", in Won Kim & Hyung-Jin Choi, ed., 'ICUIMC' , ACM, pages 570-574, 2008

[16] Gunther C. and van der Aalst W.M.P., "A Generic Import Framework for Process Event Logs", In J. Eder and S. Dustdar, editors, Business Process Management Workshops, Workshop on Business Process Intelligence (BPI 2006), volume 4103 of Lecture Notes in Computer Science, pages 81-92. Springer-Verlag, Berlin, 2006

[17] Gunther C.W. and van der Aalst W.M.P., "Fuzzy Mining: Adaptive Process Simplification Based on Multi-perspective Metrics" , In G. Alonso, P. Dadam, and M. Rosemann, editors, International Conference on Business Process Management (BPM

2007), volume 4714 of Lecture Notes in Computer Science, pages 328-343, Springer-Verlag, Berlin, 2007
[18] Gunther C.W., "XES Standard Definition", Fluxicon Process Laboratories, November 2009
[19] Hammer M., Champy J., "Reengineering the Corporation: A manifesto for Business Revolutuon", New York, 1993
[20] de Medeiros A.K.A., van Dongen B.F., van der Aalst W.M.P., Weijters A.J.M.M., "Process Mining: Extending the ?-algorithm to Mine Short Loops", BETA Working Paper Series, WP 113, Eindhoven University of Technology, Eindhoven, 2004
[21] de Medeiros, A. K. A., Genetic Process Mining. PhD thesis, Technische Universiteit Eindhoven, Eindhoven, The Netherlands, 2006
[22] Orlicky J.A., "Structuring the bill of materials for mrp", Production and Inventory Management, pages 19-42, 1972
[23] Petri C.A. , "Kommunikation mit Automaten", PhD thesis, Institut fur instrumentelle Mathematik, 1962
[24] Petruşel R., Aggregating individual models of decision-making processes. InProceedings of the 24th international conference on Advanced Information Systems Engineering (CAiSE'12), Jolita Ralyté, Xavier Franch, Sjaak Brinkkemper, and Stanislaw Wrycza (Eds.). Springer-Verlag, Berlin, Heidelberg, 47-63, 2012
[25] Reisig W., "Petri Nets: An Introduction", volume 4 of Monographs in Theoretical Computer Science: An EATCS Series. Springer-Verlag, Berlin, 1985
[26] Riehle D. and Züllighoven H., "Understanding and Using Patterns in Software Development", Theory and Practice of Object Systems, 2(1):3-13, 1996.
[27] Rozinat A., and van der AalstW. M. P., Conformance checking of processes based on monitoring real behavior. Information Systems, 33(1):64-95, 2008
[28] Sadiq S.W., Orlowska M.E., Sadiq W., Foulger C., "Data-flow and Validation in Workflow Modelling", In Fifteenth Australasian Database Conference (ADC), Dunedin, New Zealand, volume 27 of CRPIT, pages 207-214, Australian Computer Society, 2004
[29] Sadiq W., Orlowska M.E., "Analyzing Process Models using Graph Reduction Techniques", Information Systems, 25(2):117-134, 2000
[30] Sun S. X., Zhao J.L., "Activity Relations: A Dataflow Approach to Workflow Design", proceedings of International Conference on Information Systems 2006, Milwaukee, Wisconsin, USA, Paper 44, 2006
[31] Sun S.X., Zhao J.L., Nunamaker J.F., Liu Sheng O.R., "Formulating the Data-flow Perspective for Business Process Management", Information Systems Research, 17(4), pages 374-391, 2006
[32] Trcka N., van der Aalst W.M.P. , and Sidorova N., "Data-Flow Anti-Patterns: Discovering Data-Flow Errors in Workflows", In P. van Eck, J. Gordijn, , and R. Wieringa, editors, Advanced Information Systems Engineering, Proceedings of the 21st International Conference on Advanced Information Systems Engineering (CAiSE'09), volume 5565 of Lecture Notes in Computer Science, pages 425-439. Springer-Verlag, Berlin, 2009
[33] Vanderfeesten I., "Product-Based Design and Support of Workflow Processes", Eindhoven University of Technology, Eindhoven, 2009
[34] Verbeek E., van der Aalst W. M. P., "Woflan 2.0: a Petri-net-based workflow diagnosis tool", In Proceedings of the 21st international conference on Application and theory of petri nets (ICATPN'00), Mogens Nielsen and Dan Simpson (Eds.), Springer-Verlag, Berlin, Heidelberg, pages 475-484, 2000
[35] Verbeek H., Buijs J. C. A. M., Dongen B. F., and van der Aalst W. M. P., "XES, XESame, and ProM 6", Information Systems Evolution, pages. 60-75, 2011

[36] Verbeek H.M.W., Basten T., van der Aalst W.M.P., "Diagnosing Workflow Processes using Woflan", Computing Science Report 99/02, Eindhoven University of Technology, Eindhoven, The Netherlands, 1999
[37] Wang J., Kumar A., "A framework for document-driven workflow systems", In Proceedings of the 3rd International Conference on Business Process Management. Lecture Notes in Computer Science, vol. 3649, Springer Verlag, pages 285-301, 2005
[38] Weijters A.J.M.M. and van der Aalst W.M.P., "Rediscovering Workflow Models from Event-Based Data using Little Thumb", Integrated Computer-Aided Engineering, 10(2):151-162, 2003