Business Information Systems Research Center
Centrul de cercetări in Informatică Economică

Proiect PN-III-P2-2.1-PED-2016-1140

ENTERKNOW:

Aplicație Enterprise-Aware bazată pe reprezentarea hibridă și formală a cunoștințelor întreprinderii

Robert Andrei Buchmann (Director proiect), **Ana-Maria Ghiran, Mihai Cinpoeru, Alisa Harkai**

Rezumat Etapa 2018

Prezență on-line:

http://austria.omilab.org/psm/content/enterknow/info

(în portalul comunității OMiLAB - Open Models Initiative Laboratory)

http://enterknow.granturi.ubbcluj.ro

1. NOI CONSIDERAŢII PE MARGINEA PROBLEMEI ABORDATE

Etapa 2018 a vizat implementarea celei de-a doua iteraţii a demonstratorului, aducând unele extensii în toate componentele acestuia – limbajul de modelare introdus în Etapa 2017, baza de cunoştinţe, aplicaţia front-end. S-a rafinat arhitectura la un nivel de granularitate mai precis, s-a rafinat şi extins metamodelul limbajului de modelare, s-au realizat evaluări şi s-au inclus unele puncte de extensie pentru posibile dezvoltări în cercetări şi experimente viitoare.

După cum s-a prefigurat în raportul technico-ştiinţific pentru etapa 2017, tipul de aplicaţie implementată în demonstrator este un sistem de suport pentru execuţia proceselor de afaceri, cu un modul de modelare a proceselor şi unul de atribute a sarcinilor, care a fost extins în iteraţia curentă cu elemente de localizare şi inferenţe geografice bazate pe ontologia şi funcţiile GeoSPARQL (OGC, 2018).

Sistemele ce oferă suport pentru gestiunea proceselor de afaceri sunt în mod inerent ghidate de modele, folosindu-se de baze de procese interpretabile de către maşini, procesele fiind reprezentate cu ajutorul a diverse standarde – BPMN (OMG, 2018) sau EPC (Scheer A. W., 2002) pentru vizualizări grafice, XPDL (WfMC, 2018) sau chiar baze de date SQL pentru stocare. Pe de altă parte, descrierea fluxurilor de lucru prin standarde este în ultima vreme confruntată cu o cerinţă a agilităţii semantice (Karagiannis, 2016), cerinţă originară în domeniul Sistemele de Management al Cunoştinţelor şi în particular la sistemele informatice ce se folosesc de baze de cunoştinţe în locul bazelor de date tradiţionale. Cerinţa de agilitate semantică determină o migrare dinspre aplicaţiile de tip "blueprint-oriented" spre cele contextuale.

Diagramele de procese sunt forme de reprezentare a cunoştinţelor procedurale iar integrarea lor cu elemente ale contextului (organizaţional şi de execuţie) fac obiectul cerinţei de agilitate semantică menţionată. Contextul în care se execută procesele poate fi structurat conform perspectivelor propuse de Framework-ul Zachman (Zachman, 1987), utilizat ca ghid conceptual în cercetarea de faţă. Dintre faţetele Zachman, o importanţă aparte pentru implementarea demonstratorului acestui proiect o prezintă faţetele Why (obiectivul), How (procesele), Who (participanţii la procese) şi Where (locaţiile unde trebuie să se execute procesele). Aceste faţete sunt structurate într-un model de date ce înglobează cunoştinţe hibride folosindu-se de combinarea tehnologiilor Semantic Web cu limbajele agile de modelare grafică. Paradigma de integrare Linked Data e folosită pentru a asigura unificarea unei "baze de cunoştinţe hibride" derivată din surse eterogene capabile să îmbogăţească descrierea proceselor de lucru cu elemente relevante la momentul execuţiei. Sursele respective sunt diagramele create cu un limbaj de modelare adaptabil, date existente în sisteme moştenite şi informaţii geospaţiale. Mecanismele de unificare a acestor informaţii eterogene sunt bazele de grafuri RDF, inferenţele şi interogările aplicabile pe astfel de grafuri, precum şi o interfaţă standard de tip REST pentru interogare de grafuri prin HTTP – implementate pe serverul GraphDB (Ontotext, 2018.) folosit în demonstratorul proiectului pe post de Sistem de Gestiune a Bazei de Date.

Reprezentarea, conversia şi conectarea informaţiei în baza de cunoştinţe hibridă sunt asigurate de: (i) standardul GeoSPARQL pentru modelarea grafurilor cu informaţii geospaţiale, utilizat în tandem cu Google Maps şi Google Geocoder pentru a obţine coordonate de locaţii; (ii) un mecanism de interoperabilitate pentru conversia diagramelor în grafuri RDF (W3C, 2018); (iii) plug-in-ul OntoRefine

pentru conversia de structuri de date moştenite – tabele, JSON, XML etc. în grafuri RDF (OpenRefine Community, 2018).

Propunerea reflectă dezbateri recente din comunitatea ştiinţifică a Modelării Conceptuale, sub noţiunea-umbrelă de "Next-Generation Enterprise Modeling", dezbatere la care contribuie şi membrii proiectului prin intermediul reţelei colaborative Open Models Laboratory (OMiLAB, 2018). Interpretarea instanţiată în proiectul de faţă porneşte de la convergenţa între următoarele tendinţe:

1. **Integrarea modelelor diagramatice în ciclurile clasice de conversie a cunoştinţelor**, aşa cum au fost ele definite de disciplina Managementul Cunoştinţelor (Nonaka, 1991). Astfel de cicluri au fost analizate în detaliu de membrii proiectului în colaborare cu membri OMiLAB (Karagiannis et al., 2017) unde am arătat că, pentru a putea folosi limbajele de modelare în scenarii de achiziţie de cunoştinţe cu specific de domeniu devine obligatorie agilitatea limbajelor de modelare sugerată la punctul următor;
2. **Agilitatea la nivel de limbaje de modelare** devine o alternativă la limbajele de modelare standard, bazate pe consens global. Instrumentele necesare pentru a permite această agilitate sunt platformele de metamodelare – cea utilizată pentru demonstratorul de faţă fiind ADOxx, pusă la dispoziţie gratuit în cadrul comunităţii Open Models Laboratory, însă propunerea de faţă s-ar putea generaliza şi pentru alte platforme cu funcţii similare – MetaEdit+ (Tolvanen and Kelly, 2009) etc.;
3. **Cerinţa de integrare a cunoştinţelor reprezentate pentru consum uman** (diagrame) **cu cele reprezentate pentru procesare de către maşini** (grafuri de cunoştinţe) într-o manieră extensibilă şi pe care să se poată raţiona prin mecanisme automate, adică cerinţa de hibridizare abordată prin proiectul de faţă. Această cerinţă are în vedere nevoia de a stabili mecanisme de interoperabilitate între un mediu de modelare agilă a proceselor şi baza de date folosită de un sistem de execuţie a proceselor, mecanism ce a fost discutat în detaliu în (Karagiannis and Buchmann, 2018) şi va fi rezumat aici.

Sistemele de suport pentru execuţia proceselor/fluxurilor de lucru (BPMS) sunt ideale pentru a demonstra convergenţa aici menţionată, datorită naturii lor "model-driven" care e legată nu doar de faza de proiectare, ci şi de operarea curentă a astfel de sisteme – mai exact, interacţiunea cu modelele nu e limitată la ingineria software ci şi la utilizare.

Fig. 1 sumarizează sursele de informaţii hibridizate în baza de cunoştinţe a sistemului de tip BPMS precum şi metoda de inginerie software adoptată (reiterată pentru a doua oară în cadrul etapei de proiect 2018). Mai multe detalii asupra metodei de inginerie software au fost publicate în (Buchmann et al., 2018a).

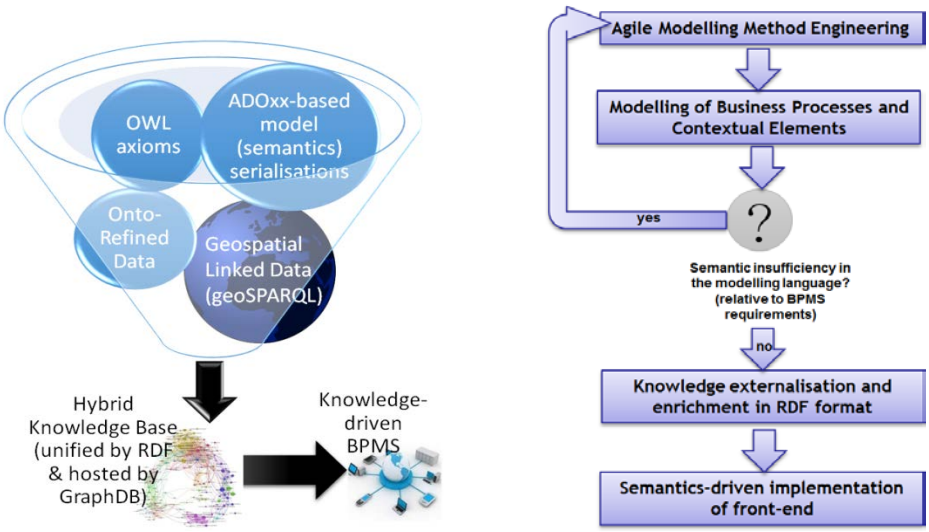

Fig. 1. Viziunea de ansamblu: sursele de informaţii hibridizate (stânga), procesul simplificat de inginerie software (dreapta)

2. METODA PROPUSĂ DE HIBRIDIZARE A CUNOȘTINȚELOR

SGBD-ului GraphDB vine cu două componente plug-in de tip "semantic lifting": (i) componenta OntoRefine pentru convertirea de date moştenite (tabele, XML, JSON) în grafuri; (ii) componenta GeoSPARQL, o ontologie şi un set de funcţii pentru raţionare geografică (comparaţii, distanţe etc). La aceste surse se adaugă cunoştinţele provenite din diagrame create cu un limbaj agil de descriere a proceselor, instrumentul de modelare oferind mecanismele necesare pentru conectarea informaţiei din diagrame la date moştenite (în cazul demonstratorului e vorba de adresele unor locaţii de interes pentru executarea de procese) şi la inferenţele GeoSPARQL (detectarea locaţiilor de interes suficient de apropiate locaţiei unde trebuie să se execute o sarcină).

Fig. 2 prezintă structura tipică de graf pe care o produce componenta OntoRefine pentru date moştenite în format tabelar - în cadrul demonstratorului s-a folosit un tabel Excel, adaptoare sunt disponibile şi pentru baze de date MySQL, vezi (D2RQ, 2018).

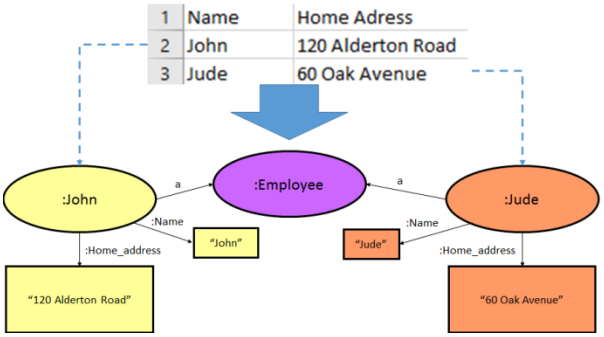

Fig. 2. Conversia structurilor tabelare în grafuri cu OntoRefine

În cadrul demonstratorului proiectului de faţă, adresele din surse externe sunt trimise serviciului Google Encoder pentru a obţine coordonatele geografice ale locaţiilor respective, coordonate stocate mai apoi

în baza de grafuri în vederea execuţei de inferenţe de tip geocomparison bazate pe standardul GeoSPARQL – în cazul de faţă e vorba de compararea locaţiilor sarcinilor declarate în modele diagramatice (locurile unde trebuie să ajungă executantul unei sarcini din proces) cu locaţii de interes (parcări, hoteluri etc.)

Fig. 3 prezintă graful corespunzător unui fragment izolat, minimalist din modele diagramatice realizate cu un limbaj agil (personalizarea se remarcă inclusiv la nivel de simboluri grafice, comparativ cu standarde precum BPMN). Fragmentul indică (i) două elemente dintr-o diagramă de proces – o decizie, o sarcină de lucru şi condiţia care face ca decizia să fie urmată de sarcină; (ii) un element instanţă dintr-o organigramă şi (iii) un hyperlink care stabileşte relaţia de responsabilitate între sarcină şi actorul (sau rolul) alocat acesteia pentru execuţie.

Derivarea grafului foloseşte ca input structura de stocare a diagramelor în ADOxx (SQL sau XML) şi implementează următoarele reguli de transformare, folosind o schemă RDF improvizată pentru diverse şabloane diagramatice. Graful derivat este expresia unui set de predicate logice în conformitate cu formalismul care stă la baza formatului RDF. Exemplul din figură poate fi exprimat la nivel de predicate astfel (considerând un AND logic pentru a agrega toate predicatele într-o bază de cunoştinţe; de asemenea am descompus relaţia n-ară _:x în predicate binare):

Predicate unare:
Decision(DoesTailorHaveRequiredTextile)
Task(PlaceOrderForTextile)
Employee(Jane)
Sequence(_:x)
Predicate binare:
Responsible(PlaceOrderForTextile,Jane)
Condition(_:x,"yes")
From(_:x,DoesTailorHaveRequiredTextile)
Condition(_:x,PlaceOrderForTextile)
Invarianţi/Axiome:
Decision(x) -> NodeSymbol(x)
Sequence(x) -> ConnectorWithAttributes(x)
Helper(from)
Helper(to)
Hyperlink(Responsible)
EditableAttribute(Condition)

Notă: în implementare, aceste predicate şi noduri au denumiri mai puţin lizibile, conforme cu schema de identificare URI şi mecanismul de generare de identificatori din ADOxx.

Stocarea acestor predicate în GraphDB permite o serie de şabloane inferenţiale (Karagiannis and Buchmann, 2018) de ale căror rezultate va beneficia orice aplicaţie client care interoghează respectiva bază de cunoştinţe. Ilustrăm în continuare o parte din aceste şabloane.

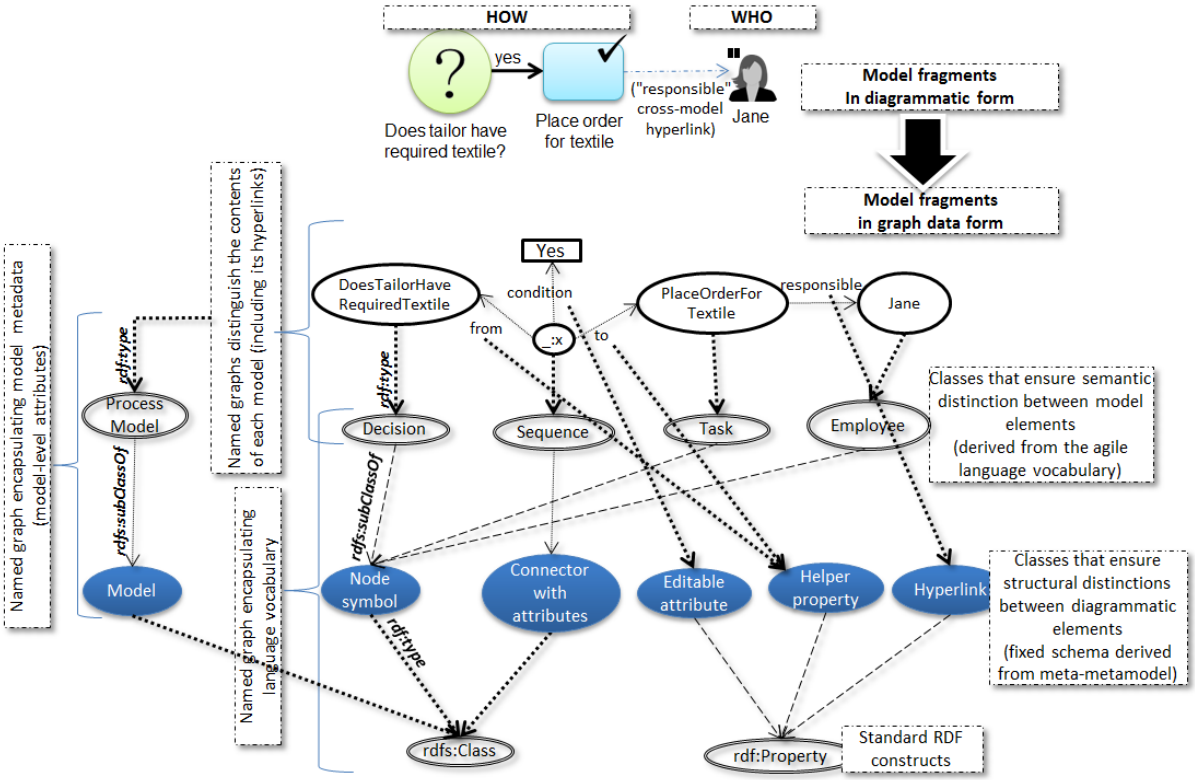

Fig. 3. Conversia structurilor diagramatice în grafuri (Karagiannis and Buchmann, 2018)

Fig. 4 oferă o viziune de ansamblu asupra hibridizării cu toate sursele de date implicate:

- Un fragment de diagramă cu adnotări minimale (etichete şi numere de telefon colectate prin intermediul instrumentului de modelare);
- Date externe care nu ar avea sens să fie colectate prin intermediul instrumentului de modelare, de exemplu date moştenite disponibile în surse externe (aici caracteristici ale spaţiilor de parcare);
- Adnotări GeoSPARQL ce permit raţionarea pe îmbinarea de diagrame şi date: nodurile derivate din limbajul de modelare sunt subordonate tipurilor oferite de ontologia GeoSPARQL (geo:Feature va subsuma orice element de diagrame care are o manifestare geospaţială). Figura indică o axiomă de înlănţuire OWL care generează o relaţie :atCoordonates între o instanţă geo:Feature şi coordonatele sale (în formatul WKT). Apoi o inferenţă SPARQL generează :recommendedParking între o sarcină din proces şi spaţiile de parcare aflate pe o anumită rază faţă de locaţia unde trebuie executată sarcina. Aceasta poate fi înlănţuită mai departe (:parkingForUser) pentru a conecta direct un anume utilizator al sistemului la parcările care i se recomandă.

Coordonatele geografice pot fi adnotate fie manual la elemente cheie din modele, fie solicitate de la un serviciu dedicat în acest scop – în cazul demonstratorului am folosit Google Maps şi Google Geocoder pentru a marca locaţiile chiar şi dacă dispunem doar de adresele acestora.

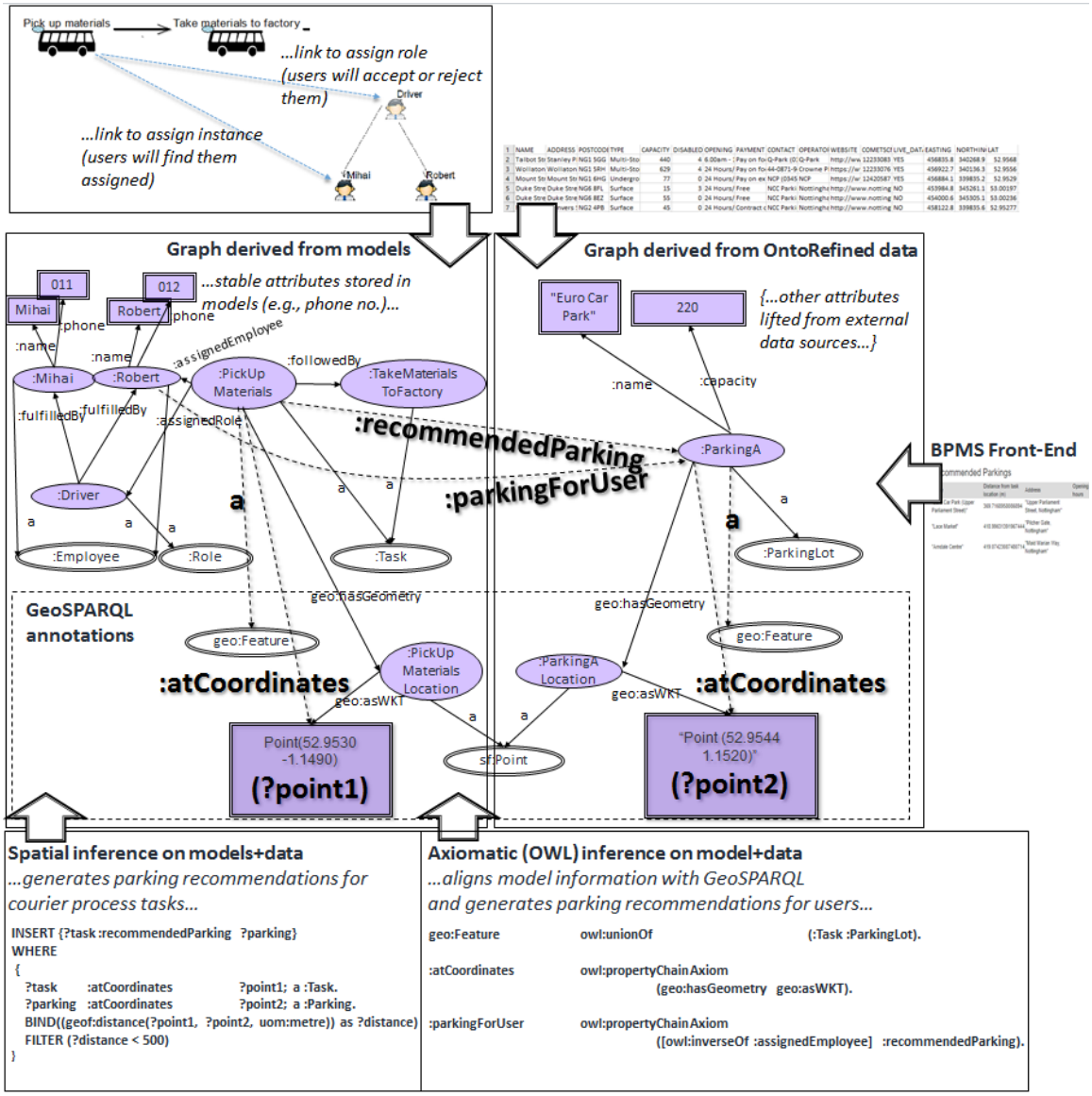

Fig. 4. Inferenţe pe reţeaua semantică obţinută din diagrame, date externe şi GeoSPARQL

3. Proiectarea şi implementarea demonstratorului

Arhitectura prezentată în raportul 2017 a fost rafinată la un nivel de granularitate ce reflectă implementarea curentă, ilustrată în Fig. 5. În aceeaşi figură evidenţiem şi cazurile de utilizare. Se poate observa o reminiscenţă de bază de date MySQL la care am apelat pentru stocarea parolelor şi conturilor de utilizatori, care nu ar trebui supuse aceloraşi principii de deschidere şi interconectare ca informaţia din GraphDB.

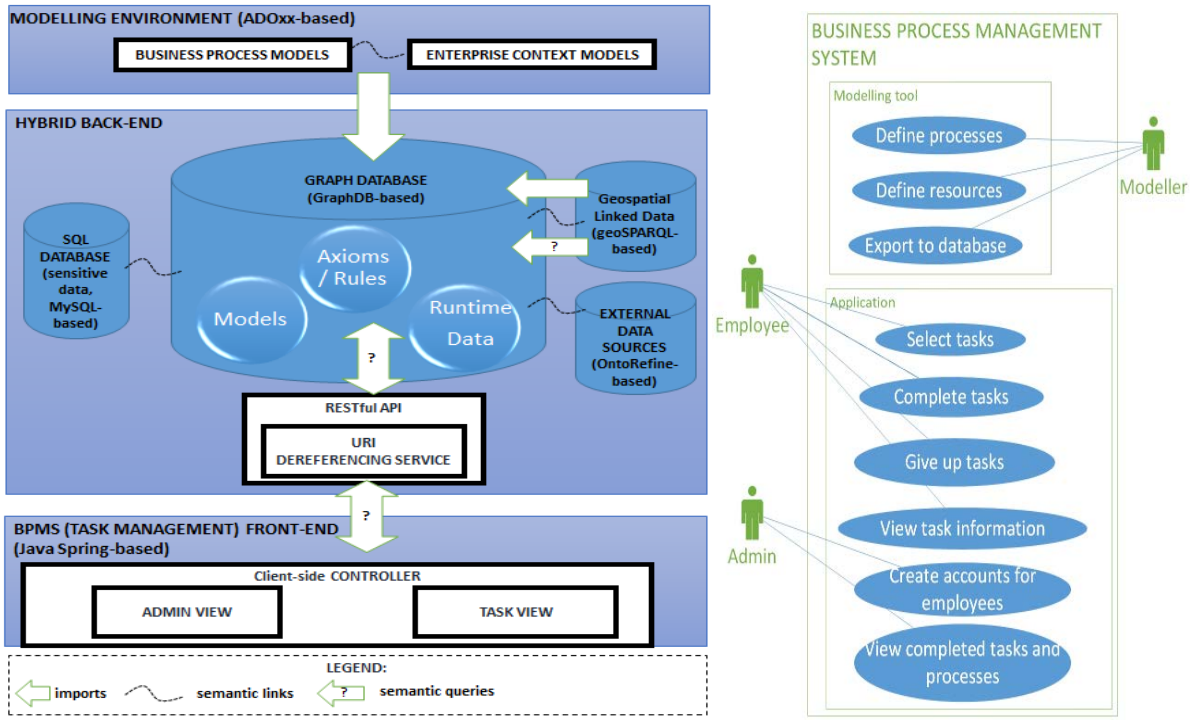

Fig. 5. Arhitectura rafinată a demonstratorului (stânga) şi cazurile de utilizare (dreapta)

Limbajul de modelare a fost proiectat în baza unui metamodel (Fig. 6) ce descompune descrierea întreprinderii în diagrame de tipuri diferite: O diagramă cu rol de "cuprins" având un format liber de tip Mind Map (Buchmann et al., 2018b), în care un obiectiv al întreprinderii este descompus radial în aspectele prevăzute de faţetele Zachman (How, Who, Where etc.). Centrul diagramei reprezintă faţeta Why (obiectivul) şi oferă un hyperlink direct la lansarea componentei front-end a demonstratorului. Elementele radiale sunt conectate la celelalte tipuri de diagrame ale limbajului – procese, locaţii, participanţi.

Implementarea modulului front-end s-a realizat în Java cu frameworkul Spring, cu Thymeleaf pentru şabloanele interfeţei cu utilizatorul şi unele componente Jquery declanşatoare de cereri spre serverul de grafuri GraphDB. Implementarea mediului de modelare s-a realizat în ADOxx. Fig. 7 arată diagrama-cuprins, o diagramă de tip Mind Map ce conectează un obiectiv la faţetele Zachman – fiecare din acestea detaliate în alte diagrame de tipuri corespunzătoare.

Fig. 8 arată un exemplu de diagramă de proces cu sarcinile adnotate cu locaţii şi responsabili. Adnotările locaţiilor sunt transmise spre Google Geocoder pentru a genera marcaje pe o hartă Google – sunt marcate locaţiile unde trebuie executat un proces şi se calculează puncte de interes suficient de apropiate (ex., parcări). Adresele punctelor de interes sunt preluate din sursa de date tabelară "moştenită" (tabel Excel convertit în grafuri cu ajutorul plug-in-ului OntoRefine oferit de GraphDB).

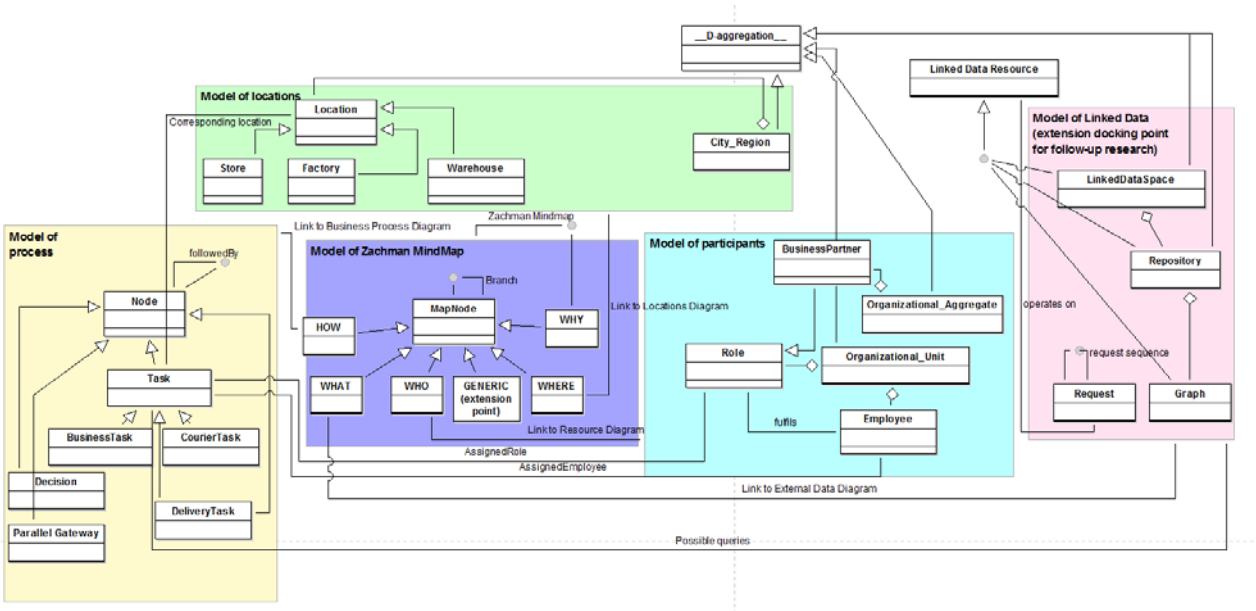

Fig. 6. Metamodelul limbajului de modelare

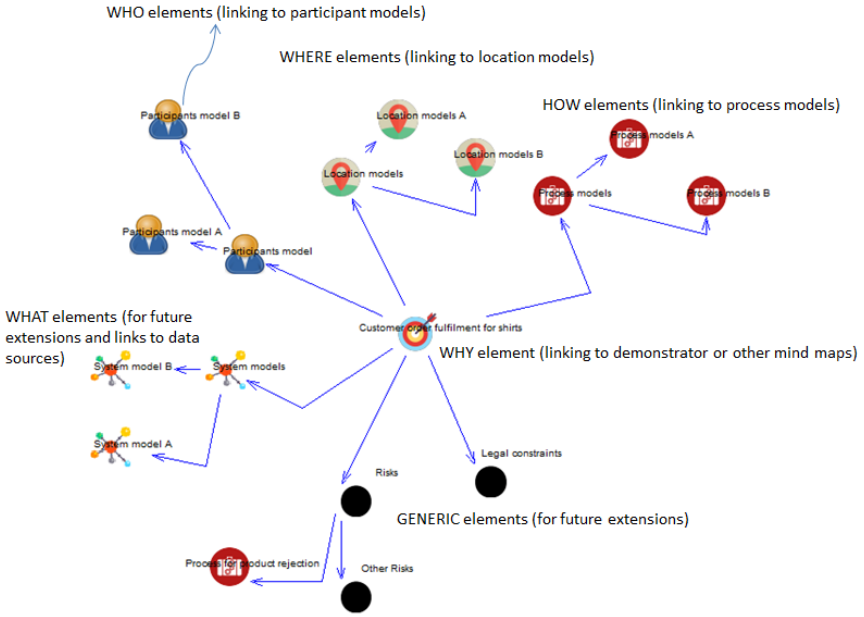

Fig. 7. Mind Map cu rol de "cuprins" pentru mediul de modelare (în dreapta variană cu notaţie personalizată pentru semantica nodurilor)

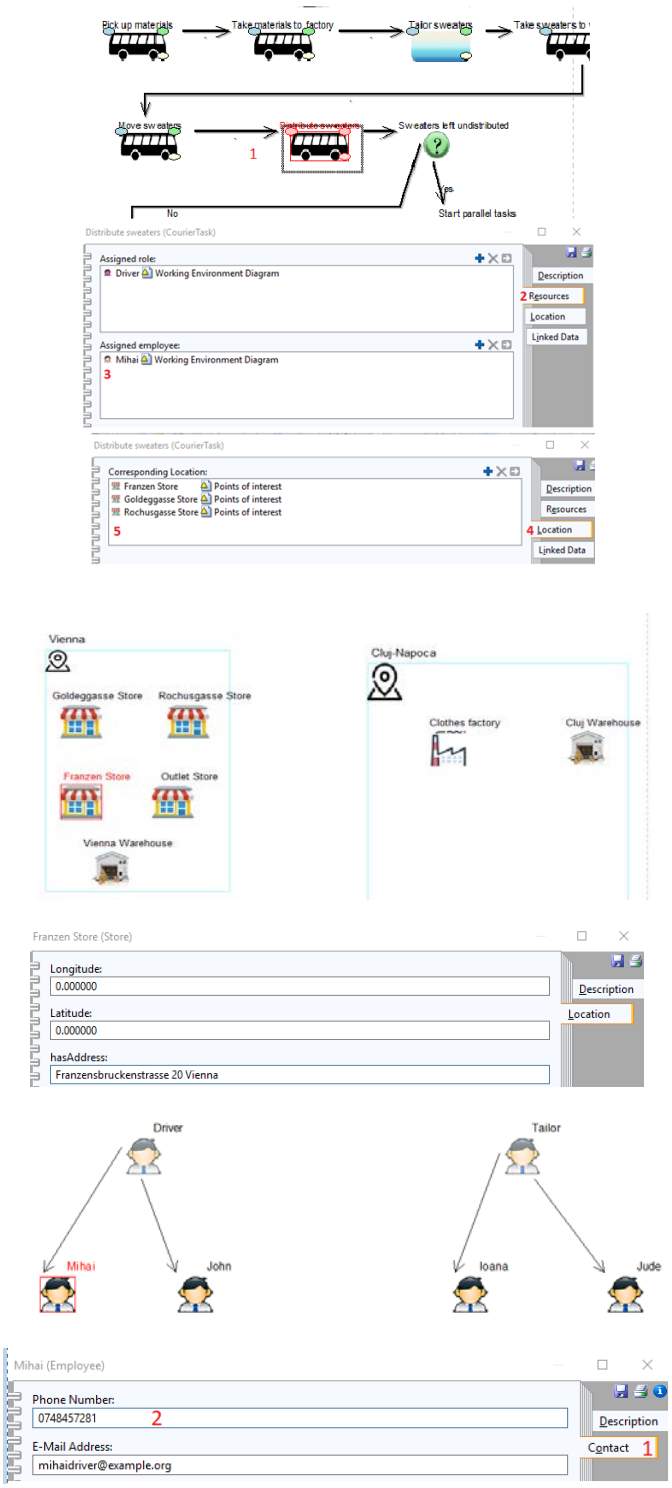

Fig. 8. Adnotarea diagramelor de proces, participanţi şi locaţii

În urma hibridizării informaţiilor, modulul front-end oferă două componente, una pentru administratori şi una pentru utilizatorii ce pot prelua sarcini de execuţie a proceselor şi au nevoie să găsească parcările

cele mai apropiate conform cu locaţiile prevăzute de modelul procesului, precum şi cu deciziile ce pot apărea pe parcurs. Deciziile sunt luate de administrator pentru întreg procesul, ceea ce determină calea de urmat prin proces în timpul execuţiei. Responsabilii cu fiecare sarcină sunt fie alocaţi la nivel de diagramă, fie îşi vor selecta sarcinile din propria consolă, caz în care în diagramă e alocat doar un rol (vezi adnotările din Fig. 8). Fig. 10 indică vizualizarea pusă la dispoziţia executantului unei sarcini, care vede într-o componentă Google Maps parcări din proximitatea locaţiei la care trebuie să îşi execute sarcina (conform informaţiei din modele).

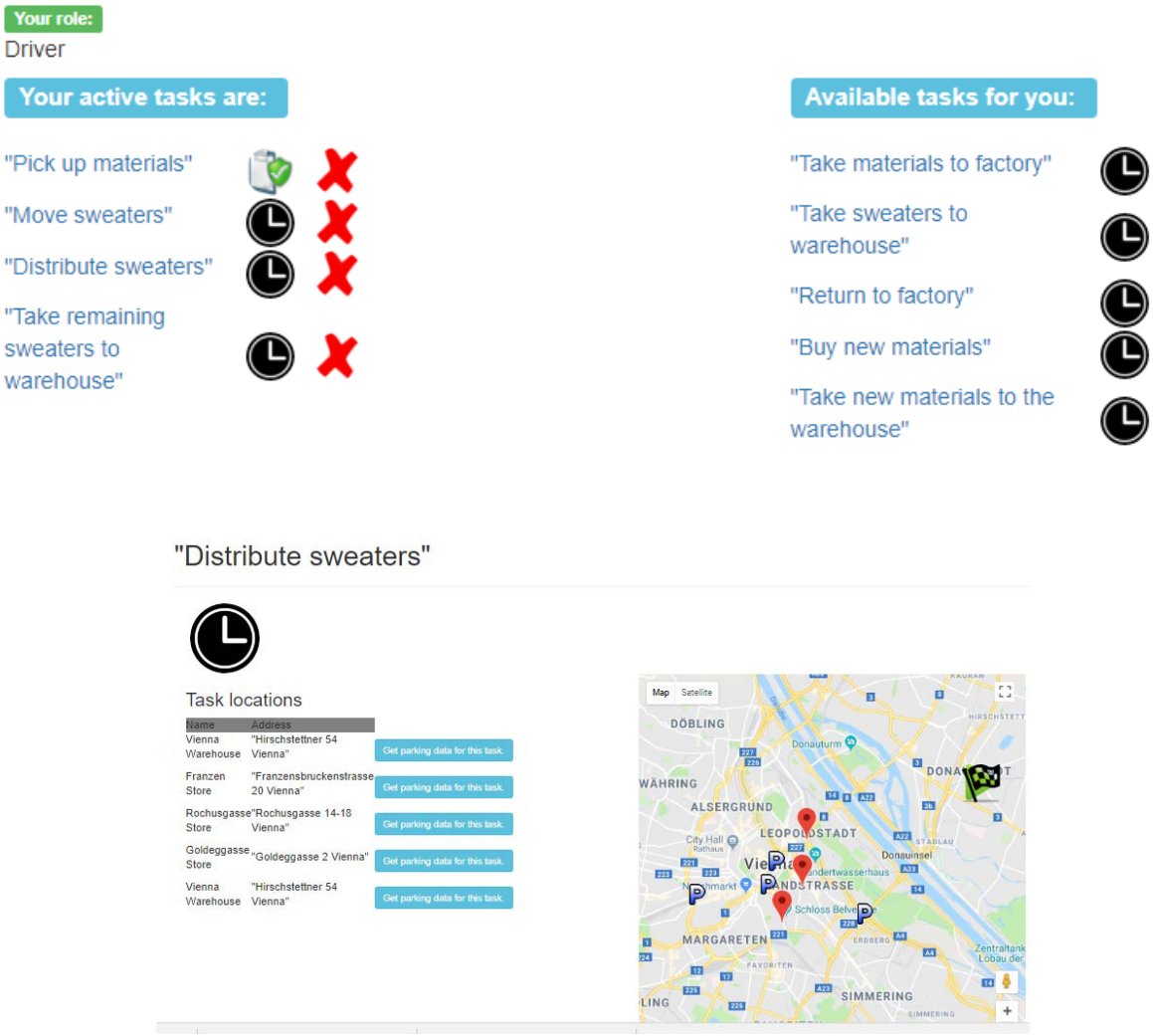

Fig. 9. Vizualizarea de către angajaţi a sarcinilor procesului cu recomandări de locaţii Google Maps

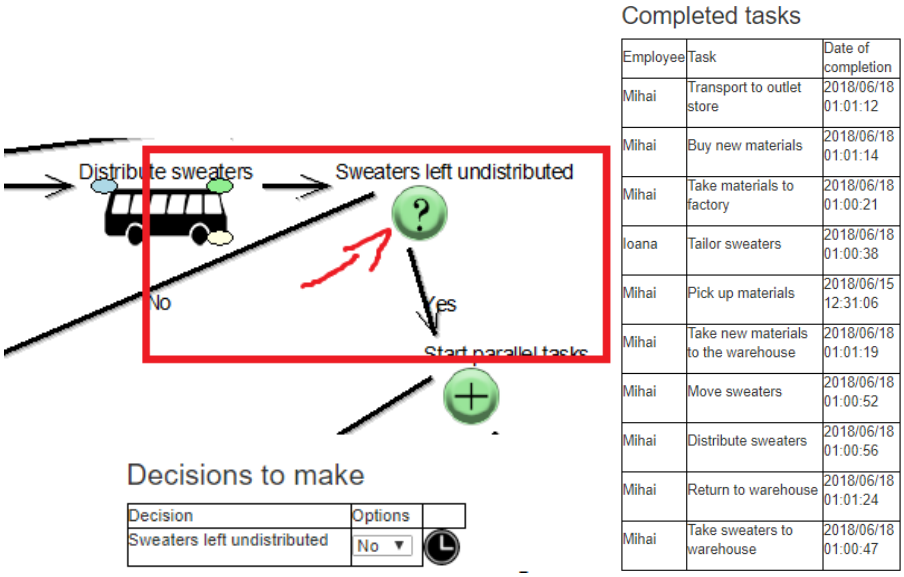

Fig. 10. Luarea deciziilor detectate în model (strânga) şi monitorizarea progresului pe pagina administratorului (dreapta)

Comparativ cu iteraţia prezentată în raportul 2017, şi în baza testelor realizate în cadrul ultimei etape s-au realizat o serie de optimizări ce au condus la cifrele indicate în Table 1: (i) obiectele îşi accesează proprietăţile doar când e nevoie de ele, evitând operaţii nenecesare. De exemplu lista tuturor sarcinilor ce succed o sarcină dată e necesară doar la momentul la care administratorul hotărăşte asupra deciziilor detectate în diagrama de proces; (ii) considerând cererile HTTP spre serviciile Google Maps, s-a urmărit economisirea acestora pentru a ne putea încadra în cota gratuită de cereri oferită începând cu revizuirea termenilor Google din iunie 2018 (aceasta a presupus o serie de teste suplimentare asupra obiectelor-locaţie). Abrevierea TSP folosită în tabel se referă la pagina de selectare a unei sarcini din proces.

Table 1. Rezultatele testelor de performanţă pentru operaţii cheie

Operaţii	Performanţă (ms)			
	Request	Response	Încarcare DOM	Total
Login (and get TSP)	596	6	243	846
Login (and get Admin Page)	1141	6	212	1360
Take task (and reload TSP)	876	7	208	1092
Get data about pending task (redirect to contact page)	59	5	195	260
Complete task (and reload TSP)	1979	6	626	2613
Load task page	247	9	514	771
Make decision	6915	5	678	7600

5. DISEMINARE

Articole în conferinţe clasa A şi workshopuri afiliate conferinţelor clasa A[1]:

1. Harkai A., Cinpoeru M., Buchmann RA.(2018) "The "What" Facet of the Zachman Framework – a Linked Data-driven Interpretation", In Proceedings of CAISE Workshops 2018, Tallinn, Estonia 2018, LNBIP 316, (pp. 197-208), ISBN 978-3-319-92897-5, Springer
2. Karagiannis D., Buchmann RA., (2018) "A Proposal for Deploying Hybrid Knowledge Bases: the ADOxx-to-GraphDB Interoperability Case", In Proceedings of the 51st Hawaii International Conference on System Sciences (HICSS), January 3-6, 2018, Hawaii, USA, (pp. 4055-4064), ISBN 978-0-9981331-1-9, University of Hawaii

Articole în conferinţe clasa B şi workshopuri afiliate conferinţelor clasa B:

3. Buchmann RA., Ghiran AM., Osman CC., Karagiannis D. (2018) "Streamlining Semantics from Requirements to Implementation Through Agile Mind Mapping Methods", In: Proceedings of Int. Conf. on Requirements Engineering: Foundation for Software Quality. REFSQ 2018, LNCS 10753, (pp. 335-351), ISBN 978-3-319-77243-1, Springer
4. Ghiran AM., Buchmann RA., Osman CC. (2018) "Security Requirements Elicitation from Engineering Governance, Risk Management and Compliance", In: Proceedings of Int. Conf. on Requirements Engineering: Foundation for Software Quality. REFSQ 2018, LNCS 10753, (pp. 283-289), ISBN 978-3-319-77243-1, Springer
5. Buchmann RA., Cinpoeru M., Harkai A., Karagiannis, D. (2018) "Model-Aware Software Engineering - A Knowledge-based Approach to Model-Driven Software Engineering", In Proceedings of the 13th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2018), March 23-24, Funchal, Madeira, Portugal, 2018 (pp. 233-240), ISBN: 978-989-758-300-1, SCITEPress
6. Mocean L.,Harkai A.(2018) "A Modeling Language for Mind Mapping in a Project Management Context", In Proceedings of the 31st International Business Information Management Association (IBIMA) Conference, April 25-26, Milan, Italy, 2018 (pp. 3233-3244), ISBN: 978-0-9998551-0-2
7. Ghiran AM., Buchmann RA., Karagiannis, D. (2018) "Towards a Framework of Techniques for Enabling Semantics-Driven Secondary Notation in Conceptual Models", In Proceedings of the 12th IEEE International Conference on Research Challenges in Information Science (RCIS 2018), May 29-31, Nantes, France, 2018, IEEE (in press)
8. Ghiran AM., Buchmann RA. (2018) "Semantic Integration of Security Knowledge Sources", In Proceedings of the 12th IEEE International Conference on Research Challenges in Information Science (RCIS 2018), May 29-31, Nantes, France, 2018, IEEE (in press)

Articole în alte conferinţe şi workshopuri internaţionale:

9. Harkai A., Cinpoeru M., Buchmann RA. (2018) "Repurposing Zachman Framework Principles for Enterprise Model-Driven Engineering.", In Proceedings of the 20th International Conference on Enterprise Information Systems (ICEIS 2018), Volume 2, March 21-24, Funchal, Madeira, Portugal, 2018, (pp. 682-689), ISBN: 978-989-758-298-1, SCITEPress.
10. Cinpoeru M.,Harkai A., Buchmann RA.(2018) "Using semantically enriched geographic representations in a business oriented scenario", In Proceedings of the 16th International Conference on Informatics in Economy (IE), May 17th, Iasi, Romania, 2018 (pp. 201-206), ISSN: 2284-7472, ASE.

Articole în reviste indexate BDI (Scopus, DBLP, ACM):

[1] conform CORE ranking 2018 (http://portal.core.edu.au/conf-ranks/), clasificare de referinţă pentru conferinţele de informatică din întreaga lume

11. Buchmann RA., Karagiannis D. (2017) "A Semantic Approach to Deploying Product-Service Systems", International Journal of Information System Modeling and Design (IJISMD) 8(1), IGI Global, DOI: 10.4018/IJISMD.2017010102
12. **În curs de evaluare (articolul e încărcat în platforma de raportare)**: Buchmann RA., Karagiannis D., Ghiran AM., Cinpoeru M., Harkai A., A semantic approach to model-driven collaboration in business process management systems, trimis la revista ACM Transactions on Social Computing
13. **În curs de evaluare (articolul e încărcat în platforma de raportare):** Bork D., Buchmann RA., Karagiannis D., Lee M., Miron ET., An open platform for modeling method conceptualization: the OMiLAB digital ecosystem, trimis la revista Communications of the AIS

Cărţi la edituri naţionale:

14. Buchmann RA., Ghiran AM. (2018), "Tehnologii semantice", Cluj-Napoca, Presa Universitara Clujeana, ISBN 978-606-37-0374-4.

Lucrări disertaţie pentru masteranzii implicaţi cu normă întreagă în proiect (susținere în iulie 2018):

15. Mihai Cinpoeru, Disertația A Hybrid Knowledge-based approach to Business Process Management Systems, la Facultatea de Matematică-Informatică, programul de studii "Sisteme Distribuite în Internet", Universitatea Babeș-Bolyai
16. Alisa Harkai, Disertația Proof-of-Concept for a Technology-Specific Modelling Language, la Facultatea de Științe Economice și Gestiunea Afacerilor, programul de studii "Business Modeling and Distributed Computing" (în engleză), Universitatea Babeș-Bolyai

Bibliografie

Buchmann RA., Cinpoeru M., Harkai A., Karagiannis, D. (2018a). Model-Aware Software Engineering - A Knowledge-based Approach to Model-Driven Software Engineering, In *Proceedings of the 13th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2018*), Funchal, Madeira, Portugal, (pp. 233-240), SciTe Press.

Buchmann, R. A., Ghiran, A. M., Osman, C. C., & Karagiannis, D. (2018b). Streamlining Semantics from Requirements to Implementation Through Agile Mind Mapping Methods. In *International Working Conference on Requirements Engineering: Foundation for Software Quality* (pp. 335-351), LNBIP 273, Springer.

D2RQ, 2018. *official website*. http://d2rq.org/, accesat Iun. 2018

Karagiannis, D. (2016). Conceptual Modelling Methods: The AMME Agile Engineering Approach. In *International Conference on Informatics in Economy* (pp. 3-19). Springer.

Karagiannis D, Buchmann RA, Walch M, 2017. *How can diagrammatic conceptual modelling support Knowledge Management?* Proceedings of ECIS 2017, paper 101, Association for Information Systems

Karagiannis D., Buchmann RA., (2018). A Proposal for Deploying Hybrid Knowledge Bases: the ADOxx-to-GraphDB Interoperability Case, In *Proceedings of the 51st Hawaii International Conference on System Sciences (HICSS)*, Hawaii, USA, (pp. 4055-4064), Univ. Hawaii.

Nonaka, I. (1991). The knowledge-creating company, *Harvard Business Review* 69, (pp. 96-104).

OGC, 2018. *The GeoSPARQL standard*, http://www.opengeospatial.org/standards/geosparql, accesat Iun. 2018

OMG, 2018. Process Model and Notation BPMN v2.0 *official website*, http://www.bpmn.org/ accesat Iun 2018

OMiLAB, 2018. *Open Models Laboratory - the official project page*, http://www.omilab.org/psm/home, accesat Iun. 2018

Ontotext, 2018. *GraphDB official website*, http://graphdb.ontotext.com, accesat Iun. 2018

OpenRefine Community, 2018. *OpenRefine – the official project page*, http://openrefine.org, accesat Iun. 2018

Scheer, A.W. , 2002. ARIS. *Vom Geschäftsprozess zum Anwendungssystem.* Springer, (p. 20).

Tolvanen, J. P., & Kelly, S., 2009 . MetaEdit+: defining and using integrated domain-specific modeling languages, in *Proceedings of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications*, New York, NY, USA, (pp. 819–820).

W3C, 2018. *RDF 1.1 Concepts and Abstract Syntax*, http://www.w3.org/TR/rdf11-concepts, accesat Iun. 2018

WfMC, 2018. XML Process Definition Language (XPDL) *official website*, http://www.xpdl.org/, accesat Iun. 2018

Zachman J A, 1987, *A framework for information systems architecture*. IBM Systems Journal 26 (3): (pp. 276-292).