Dynamic process workflow routing using Deep Learning

Kareem Amin[1, 3] Stelios Kapetanakis[4], Klaus-Dieter Althoff [1, 2], Andreas Dengel[1,3], Miltos Petridis[5]

[1] German Research Center for Artificial Intelligence, Smart Data and Knowledge Services, Trippstadter Strasse 122, 67663 Kaiserslautern, Germany
{ kareem.amin, klaus-dieter.althoff, andreas.dengel } @dfki.uni-kl.de
[2] Institute of Computer Science, Intelligent Information Systems Lab, University of Hildesheim, Hildesheim, Germany,
[3] Kaiserslautern University, P.O. Box 3049, 67663 Kaiserslautern, Germany
[4] School of Computing Engineering and Mathematics, University of Brighton,
s.kapetanakis@brighton.ac.uk
[5] Department of Computing, University of Middlesex, London, UK
m.petridis@mdx.ac.uk

Abstract. Dynamic business processes are challenged by constant changes due to unstable environments, unexpected incidents and difficult to predict behaviours. In industry areas like customer support, complex incidents can be regarded as instances of a dynamic process since there can be no static planning against their unique nature. Support engineers will work with any means at their disposal to solve any emerging case and define a custom prioritization strategy, to achieve the best possible result. To assist with this, in this paper we describe a novel workflow application to address the tasks of high solution accuracy and shorter prediction resolution time. We describe how workflows can be generated to assist experts and how our solution can scale over time to produce domain-specific reusable cases for similar problems. Our work is evaluated using data from 5000 workflows from the automotive industry.

Keywords: Business Processes, Case-based Reasoning, Deep Learning, Natural Language Processing.

1 Introduction

Customer support is the most important service of a business. Speaking from a customer's experience there are only a few times where customer service was great, without the feeling of treating her/him as just another ticket. Effective customer support poses several challenges for a company since any solitary case can involve a variety of complex factors as well as substantial obscurity and uncertainty in its description. For a trained engineer to complete a series of workflow cases with all cases must be prioritized and executed on a daily schedule. Routing must be determined based on each case problem definition, complexity, priority in accordance with any historical evidence (past workflow cases) that may lead to a solution. This work proposes a hybrid solution using deep learning and case-based reasoning (CBR) on business process workflows to increase solution accuracy and minimize the cost of a solution retrieval.

The growth of intensive data-driven decision-making has caused broad recognition [1], and the promise that Artificial Intelligence (AI) technologies can augment it even further. Within the Case-based Reasoning community there have been several examples of applying data-driven methods to fast changing work environments with several benefits from it. Recently, the customer experience industry has adopted a data-centric vision in an equivalent way, as companies embrace the power of data to optimize their business workflows and the quality of their services [1].

This work focuses on large-scale customer support, helping help-desk managers to optimize their prioritization strategies and achieve increased performance. A key concept in that is timely case resolution, measured in resolved cases per minute, which usually leads to high resolution vs. lower accuracy. Research on successful customer support ticket resolutions has identified several features that influence resolutions results. For example, the work of Maddern et al. [14] looks at the effect of grammatically incorrect sentences, abbreviations, mixes among different languages and semantic challenges. Besides the knowledge containers domain vocabulary: how similarity measures are formulated and can identify the adaptation knowledge [7].

Customer support cases usually resemble an application workflow which follows a certain business process. Business processes can be represented sets of activities with temporal relationships and constraints. Business processes are highly standardized to be monitored automatically [31] [32] [33]. Several standards are now available to that can be integrated with bespoke large scale portals. The Business Process Modelling Notation (BPMN) developed by the Business Process Management Initiative (BPMI) and Object Management Group (OMG) provides a standard for the graphical representation of workflow-based business processes [28]. Standards produced for business process representation aim to cover the definition, orchestration and choreography of a business process. Over the last few years, several standards have emerged and are widely accepted and supported by mainly Service Oriented Architecture (SOA) enterprise technologies. An example is the OASIS Business Process Execution Language (BPEL), short for Web Services BPEL (WS-BPEL) [29] and the XML Process Definition Language (XPDL) which is standardized to interchange Business Process definitions between different workflow products and systems [30].

Deep Learning algorithms are effective when dealing with learning from substantial amounts of both structured and unstructured data. Within the CBR paradigm, Deep Learning models can benefit from any available data, any integration of the two faces substantial challenges [3]. While Deep Learning can be applied to learn from large volumes of labeled data, it can also be attractive for learning from substantial amounts of unlabeled/unsupervised data [4][5][6], making it attractive for extracting meaningful representations and patterns from large volumes of Workflow Data.

This paper proposes a hybrid approach using CBR and Deep Learning to mitigate the challenges that come from complex workflow domains. This approach is being evaluated with Help-Desk support engineers while prioritizing and solving new, raw-content workflows. We present DeepTMS, a hybrid Textual Case-based reasoning (hTCBR) approach using Deep Neural Networks while a) not relying on manually constructed similarity measures as with traditional CBR and b) it does not require domain expertise to decode any domain knowledge.

This paper is structured as follows: First we describe the related work to our approach. Section 3 explains our approach, our domain challenges and the followed solution architecture. Section 4 presents the carried-out evaluation with domain experts to ensure the efficiency of our proposed approach. Finally section 5 concludes this work and presents our future directions.

2 Related Work

A business process is tightly dependent on its workflow representation. When monitoring information about a business process, the current workflow state must be analysed and compared using domain/model knowledge and knowledge gained from experience. As problems usually recur, if similar cases are found this can provide the context for reasoning about the workflow or, if no such precedent can be found, new knowledge can be derived in the form of a new case that can be stored in the system for later use. This approach matches the behaviour and process of Case-Based Reasoning (CBR) systems which follows the Retrieve, Reuse, Revise, Retain model [2]. CBR seems an effective way of monitoring business processes [31][32] when represented as graphs and spatio-temporal [34] or structural similarity measures are applied [27].

Related work to this research also relates to text processing with mixed languages, customer support and CBR systems and automation of text relation extraction. Textual CBR supports cases represented as text. Text representation states several challenges since text is unstructured and can have grammatically incorrect sentences. This research can be compared to the work presented in [19] [20] [21], where hybrid CBR approaches (CBR with Natural Language Processing – NLP-) frameworks were used to process the knowledge written in free text. In this work, NLP frameworks were not able to process text spanned across different languages since there were several issues related to accurate sentence parsing. Therefore, we suggest a different approach using Deep Neural Networks to ease the task of finding similarities between workflows and automate the knowledge from textual workflows.

HOMER [22] [23] is a help desk support system designer for the same purpose of DeepTMS. HOMER used an object-oriented approach to represent cases and used a question-answering approach to retrieve cases. HOMER showed very good results when it first presented in 1998 and after its further improvement in 2004. However, any existing fast-pace work environments demand solutions that can deal with big amounts of data in real time with minimum human interference. Comparing to DeepTMS, we focused more on how to automate the extraction of similarities and deal with unstructured or mixed-languages text, but this approach also can't be automated to be integrated in business environments.

Finding the relation between text and extract features are key criteria in the success of any textual CBR system. These tasks require a lot of effort and normally can take a long time to be done accurately. Different approaches have been presented to build text similarities and find higher order relationships [24]. The work of automating knowledge extraction using Neural Networks can be compared to the work presented in [25] where authors represented the text using dubbed Text Reasoning Relevant work has been seen in Graph (TRG), a graph- based representation with expressive power to represent the chain of reasoning underlying the analysis as well as facilitate the adaptation of a past analysis to a new problem. The atuhors have used manually constructed lexico-syntactic patterns developed by Khoo [26] to extract the relations between texts.

3 Hybrid Textual CBR Approach on Workflows

Text is used to express knowledge. Text is a collection of words in any well- known language that can convey a meaning (i.e., ideas) when interpreted in aggregation [8]. To build a textual CBR system we follow the system process and how normally the workflow experts prioritize and route cases. From this process key attributes are identified as key ones to decide. In a support ticket management system these can be:

1. Subject
2. Content
3. Sender Group (The company was organized internally in diverse groups and each group had its own applications and systems)
4. Case priority assigned as assigned by the team who reported it.

Upon the above attributes, an expert can decide how to proceed with this workflow and the CBR case can be defined as:

1. Case Generation: Since attributes are few, workflow cases can have flat attribute-value representation features
2. Case Retrieval: Since NLP has substantial complexity case similarities require a rich, context-aware similarity measure. As such a trained neural network for identifying and recommending solutions from the historical case base can be selected.
3. Case Adaptation: Adaptation rules can be generated based on agent behavioral patterns and be recorded in the case management workflow “memory” of the CBR system

NLP challenges in Business process workflows include tedious and time-consuming building cases task for domain experts since they are not able to cope with the cases numbers. Any existing knowledge base as well as new tickets can be received in a multi-lingual format (e.g. English, German, French, etc.). Multi-languages add substantial complexity in the text analysis and pre-processing both in building and retrieving similar cases. Cases can be written by non-native speakers and can contain several grammar mistakes or vague domain abbreviations. Due to the last two challenges it is not possible to resort to any traditional NLP frameworks for text understanding like TwitterNLP and Stanford NLP, since their application does not return sufficient results. Our approach is based on Deep Neural Networks and Word Embeddings to improve the text pre-processing and similarity measures. Therefore, we propose a solution architecture which connects end to end: data, the CBR process and workflow experts

DeepTMS solution architecture consists of three main modules (See Figure 1):

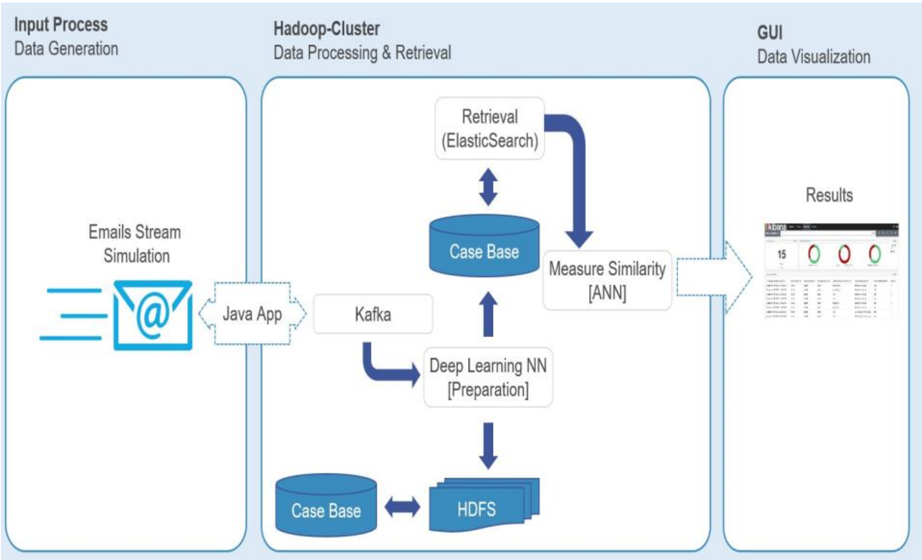

Fig. 1. DeepTMS Solution Architecture

1. Input Process (Data Generation) Module: This module is responsible for generating and simulating the emails (tickets) stream.
2. Map/Reduce -Hadoop- Cluster (Data Processing & Retrieval): This mod- ule is responsible for receiving the tickets and doing the ticket content pre- processing/processing, then retrieve the similar tickets from the Case Base (Case Generation, Retrieval & Retain).
3. Graphical User Interface (Data Visualization): This module is responsible for visualizing the results to the system end-users.

Our proposed architecture combines a Deep Neural Network with CBR to capture and decode domain knowledge in the context of NLP. It is applied throughout the task of prioritizing cases based on their content and it measures text similarity based on their semantics. We present several Neural Network types to represent a sequence of sentences as a convenient input for our different models.

The proposed methodology supports several NLP modules to handle workflow cases. These can include Support Vector Machines (SVM) and/or Vectorization to prioritize cases. Large volumes of case can be used to test the methodology as well as several states of the art neural network models like: Convolutional Neural Networks (CNNs), Recurrent Neural Network (RNNs), and Long Short- Term Memory (LSTMs) [16] to test and compare results

Vocabulary

Vocabulary is one of the knowledge containers that represents information collected from the domain to express knowledge [17]. By filling in this container we identify terms that are useful for the main system tasks. The acquisition of the domain vocabulary can have direct effects on any system performance, and that's why it is usually done with intensive help from domain experts. To improve any domain acquired vocabulary, we can follow the typical three methods described in [7]. Out of domain words and extracted key features that represent certain text can be identified by using the Word2Vec models [12]. In the next section we describe how exactly Word2Vec worked to build neural word embeddings.

Word Embedding

Most of the Deep Learning models can't process strings or plain text. They require vectorized representation as inputs to perform any sort of job, classification, regression, etc. Several NLP systems and techniques treat words as atomic units, therefore, in order to apply a Deep Learning model on NLP, words are vectorized using word embeddings is the process of converting text into a numerical representation for further processing. The distinct types of word embeddings can fall into two main categories:

1. Frequency-based embedding (FBE):

FBE algorithms focus mainly on the number of occurrences for each word, which requires a lot of time to process and exhaustive memory allocation to store the co-occurrence matrix. A severe disadvantage of this approach is that quite important words may be skipped since they may not appear frequently in the text corpus.

2. Prediction-based embedding (PBE):

PBE algorithms are based on Neural Networks. These methods are prediction based in the sense that they assign probabilities to seen words. PBE algorithms seem the present state of the art for tasks like word analogies and word similarities.

PBE methodologies were known to be limited in their word representations until Mitolov et al. introduced Word2Vec to the NLP community [12]. Word2vec consists of two neural network language models: A Continuous Bag of Words (CBOW) and Skip-gram. In both models, a window of predefined length is moved along the corpus, and in each step the network is trained with the words inside the window. Whereas the CBOW model is trained to predict the word in the center of the window based on the surrounding words, the Skip-gram model is trained to predict the context based on the central word. Once the neural network has been trained, the learned linear transformation in the hidden layer is regarded as the word representation. In this work we have used Skip-gram model since it demonstrates better performance in semantic task identification [13].

Text Pre-Processing

In the text Pre-Processing stage, raw text corpus preparation tasks are taking place in anticipation of text mining or NLP. Models like Word2Vec can be trained over case corpuses to build cases used in similarity measures. As any text pre-processing tasks, two main components can be identified, these of Tokenization and Normalization. Tokenization is a step which splits longer strings of text into smaller pieces, or tokens. Normalization generally refers to a series of related tasks meant to put all text on a level playing field: converting all text to the same case (upper or lower), removing punctuation, converting numbers to their word equivalents, and so on. Normalization puts all words on equal footing and allows processing to proceed uniformly. Normalizing text can mean performing several tasks, but for our approach, we will apply normalization in four steps: 1. Stemming, 2. Lemmatization 3. Eliminating any stopping words (German or English) 4. Noise Removal (e.g. greetings & signatures). The Word2Vec model or any other model that could be built as a substitution to the traditional taxonomies.

Similarity Measures

Similarity measures are highly domain dependent and used to describe how cases are related to each other. In CBR, comparison of cases can be performed along multiple important dimensions [9] [11]. Cases that only match partially, can be adapted to a problem situation, using domain knowledge contained in the system [10]. Thus methods Information Retrieval (IR) that are based only on statistical inferences over word vectors, are not appropriate or sufficient. Instead, mechanisms for mapping textual cases onto a structured representation are required. A basic assumption for applying the principle for similarity measures is that both arguments of the measure follow the same construction process. This allows comparing the corresponding sub-objects in a systematic way. For our system we defined the two types of similarity measures: Local Similarity Measures and Global Similarity Measures. Local Similarity Measures describe the similarity between two attributes and the Global Similarity Measures describe the similarity between two complete cases.

Local Similarity Measures (LSM): LSM are heavily dependent on local domain expertise. We have mainly four attributes which are distinctive except for the email subject and content. For the Priority (integer) and Sending Groups (distinctive strings) we used distance functions. For the email subject and content, we counted upon the Word2Vec model to give us the similarity degrees between different texts, after applying all the prepossessing tasks.

Global Similarity Measures (GSM): GSM defines the relations between attributes and gives an overall weight to the retrieved case. The weight of each attribute demonstrates its importance within the case. Methods like the weighted Euclidean distance for the calculation of the global similarity can be applied, as also shown in [15] .

4 Experimental Evaluation

Our proposed methodology is being evaluated on a real helpdesk environment which offers customer support service. This work has been a joint application between the the German Research Center for Artificial Intelligence (DFKI) and a Multinational Automotive Company (the company) with multiple offices throughout the world. Inside the company, most of the helpdesk cases come through emails to a dedicated team. Once received experts prioritize the cases and assign them to specialist engineers both inside and outside the team to work on it. The company has several historical datasets describing a plethora of issues they have happened along with given solutions. A historical workflow case could be represented in the form of Problem Description, Solution and Keywords. When new tickets arrive, an expert should search within the company's knowledge base to confirm whether any solution(s) exists or not.

The system evaluation is considering the workflow priority (as provided by the neural network) and any retrieved neighbor cases and the suggested solutions to the visited case.

During our system testing and evaluation phase, we decided to use different Neural Network models to explore, validate and compare accuracy results for each model. We applied three Neural Network models: CNNs, RNNs, and LSTMs [16]. Word2Vec was applied to vectorize text input and build word representations in the vector space (See Figure 2). Sequences of such vectors were processed using various neural net architectures.

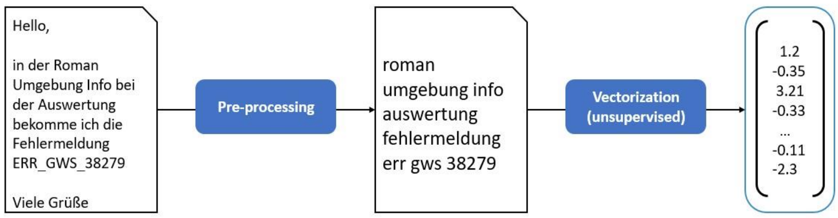

Fig. 2. Text Vectorization

Word2Vec was built using 300,000 historical workflow cases in an unsupervised training mode. All networks were built with one hidden layer, and utilized the a custom trained Word2Vec model. To train the three different neural net models, we have also used 300,000 old tickets with known priorities in a supervised learning process. An additional 10,000 tickets were used to evaluate the models in prioritizing the test tickets automatically. Table 1 summarizes the prioritizing stage results.

Table 1: Prioritization Results

Neural Network Model	**Accuracy**	**Precision**	**Recall**	**F1**
Convolutional Neural Net (CNN)	82.67%	82.52%	82.64%	82.58%
Recurrent Neural Net (**RNN)**	89.28%	89.19%	89.27%	89.23%
Long Short-Term Memory Net **(LSTM)**	92.35%	92.13%	92.23%	92.16%

Further to the initial results, a second form evaluation was conducted using the best performing algorithm on cases. The evaluation was conducted among the company experts and it was based on DeepTMS's performance on a qualitative level. DeepTMS suggested 10 solutions to any new workflow case. Experts were called to decide whether the most relevant solution was included in the retrieved corpus of cases. 4 slots were created among 10000 workflow cases and the following results were obtained:

Was it between retrieved wokflows one and three: 7764 cases – 77.64%
Was it between retrieved wokflows four and seven: 1468 cases – 14.68%
Was it between retrieved wokflows eight and ten: 692 cases – 6.92%
Was not listed: 76 cases – 0.76%

DeepTMS is using neural networks in case pre-processing to eliminate the redundant text and pass the most relevant text to deep neural networks for prioritization purposes. For our evaluation LSTM seems to out- performed all the other neural network models, however it is prone to computational overheads both during its training phase, and its text processing afterwards. CNNs seem appropriate to areas where changes occur in the network architecture and can give promising results in text processing as well [18]. CNNs are faster in training and processing phases than RNNs and LSTMs. Since an LSTM is a special RNN case they seemed to per- form well on text tasks, better t standard CNNs and worse than LSTMs. In terms of training and processing performance they take longer than CNNs and less time compared to LSTMs. Word Embedding training using Word2Vec and their utilisation within the neural networks models can give a descent performance and it can be improved with more text we use in building the model, since it expands the word corpus and improves the ability to find relationships between words. In our evaluation word2vec is built based on 50000 cases.

5 Conclusions

This paper presents a novel approach to Workflow management systems using textual CBR and Deep neural networks. Automatic feature extraction can be possible using such a hybrid technique and the results are promising as it has been seen in a real application. DeepTMS seems a solid framework to begin our work in this area and we aim to expand it towards real time case processing as well as experimenting with more advanced deep learning algorithms such as adversarial generative networks or Siamese networks. This work has shown that it is possible to have a solid, hybrid, text handling architecture using CBR however more work is required to demonstrate its general applicability.

References

1. Erik Brynjolfsson, Kristina McElheran, Data in Action: Data-Driven Decision Mak- ing in U.S. Manufacturing, Center for Economic Studies (CES), January 2016
2. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Communications 1(7) (March 1994)
3. XUE-WEN CHEN, XIAOTONG LIN, Big Data Deep Learning: Challenges and Perspectives, IEEE Access (Volume: 2), Page(s): 514 - 525,May 2014
4. Bengio Y (2013) Deep learning of representations: Looking forward. In: Proceedings of the 1st International Conference on Statistical Language and Speech Processing. SLSP13. Springer, Tarragona, Spain. pp 137.
5. Bengio Y, LeCun Y (2007) Scaling learning algorithms towards, AI. In: Bottou L, Chapelle O, DeCoste D, Weston J (eds). Large Scale Kernel Machines. MIT Press, Cambridge, MA Vol. 34. pp 321360.
6. Bengio Y, Courville A, Vincent P (2013) Representation learning: A review and new perspectives. Pattern Analysis and Machine Intelligence, IEEE Transactions on 35(8):17981828. doi:10.1109/TPAMI.2013.50
7. Richter, M.M.: Introduction. In: Lenz, M., Bartsch-Sporl, B., Burkhard, H.D., Wess, S. (eds.) Case-Based Reasoning Technology. LNCS (LNAI), vol. 1400, pp.116. Springer, Heidelberg (1998)
8. Richter, M. M., Weber, R., Springer-Verlag GmbH. (2016). Case-Based Reasoning: A Textbook.
9. Ashley, K., Modeling Legal Argument, Reasoning with Cases and Hypotheticals. MIT-Press, 1990
10. Aleven, V., Teaching Case-Based Argumentation through a Model and Examples. Ph.D. Dissertation, University of Pittsburgh, Intelligent Systems Program, 1997
11. Stefanie Brninghaus and Kevin D. Ashley (1998) How Machine Learning Can be Beneficial for Textual Case-Based Reasoning. In: Proceedings of the AAAI- 98/ICML-98 Workshop on Learning for Text Categorization (AAAI Technical Re- port WS-98-05). Pages 71-74.Madison, WI.
12. Tomas Mikolov and Kai Chen and Greg Corrado and Jeffrey Dean, Efficient Estimation of Word Representations in Vector Space, NIPS'13 Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, 2013
13. Altszyler, Edgar and Sigman, Mariano and Fernndez Slezak, Diego, Comparative study of LSA vs Word2vec embeddings in small corpora: a case study in dreams database, 2016
14. Maddern, M., Maull, R., Smart, A. (2007). Customer satisfaction and service quality in UK financial services. International Journal of Production and Operations Management, 27, 998-1019
15. Bach K., Althoff KD., Newo R., Stahl A. (2011) A Case-Based Reasoning Approach for Providing Machine Diagnosis from Service Reports. In: Ram A., Wiratunga N. (eds) Case-Based Reasoning Research and Development. ICCBR 2011. Lecture Notes in Computer Science, vol 6880. Springer, Berlin, Heidelberg
16. Hochreiter, Sepp & Schmidhuber, Jrgen. Long Short-term Memory. Neural com-

putation. 1997.

17. Richter MM, Lenz M, Bartsch-Sprl B, Burkhard H-D et al, Introduction. In: Case based reasoning technology: from foundations to applications. Lecture notes in ar- tificial intelligence, vol 1400. Springer, Berlin, p 1, 1998
18. Yoon Kim, Convolutional Neural Networks for Sentence Classification, Conference on Empirical Methods in Natural Language Processing, 2014
19. Stram R., Reuss P., Althoff KD. (2017) Weighted One Mode Projection of a Bipar- tite Graph as a Local Similarity Measure. In: Aha D., Lieber J. (eds) Case-Based Reasoning Research and Development. ICCBR 2017. Lecture Notes in Computer Science, vol 10339. Springer, Cham
20. Reuss P., Witzke C., Althoff KD. (2017) Dependency Modeling for Knowledge Maintenance in Distributed CBR Systems. In: Aha D., Lieber J. (eds) Case-Based Reasoning Research and Development. ICCBR 2017. Lecture Notes in Computer Science, vol 10339. Springer, Cham
21. Reuss P. et al. (2016) FEATURE-TAK - Framework for Extraction, Analysis, and Transformation of Unstructured Textual Aircraft Knowledge. In: Goel A., Daz- Agudo M., Roth-Berghofer T. (eds) Case-Based Reasoning Research and Develop- ment. ICCBR 2016. Lecture Notes in Computer Science, vol 9969. Springer, Cham
22. Roth-Berghofer T.R, Learning from HOMER, a Case-Based Help Desk Support System. In: Melnik G., Holz H. (eds) Advances in Learning Software Organizations. LSO 2004. Lecture Notes in Computer Science, vol 3096. Springer, Berlin, Heidel- berg
23. Gker M. et al., The development of HOMER a case-based CAD/CAM help-desk support tool. In: Smyth B., Cunningham P. (eds) Advances in Case-Based Reason- ing. EWCBR 1998. Lecture Notes in Computer Science, vol 1488. Springer, Berlin, Heidelberg
24. Ozturk, Pinar & Prasath, R.Rajendra & Moen, Hans. (2010). Distributed Repre- sentations to Detect Higher Order Term Correlations in Textual Content.
25. Sizov G., ztrk P., tyrk J. (2014) Acquisition and Reuse of Reasoning Knowledge from Tex- tual Cases for Automated Analysis. In: Lamontagne L., Plaza E. (eds) Case-Based Reason- ing Research and Development. ICCBR 2014. Lecture Notes in Computer Science, vol 8765. Springer
26. Khoo, C.S.G.: Automatic identification of causal relations in text and their use for improving precision in information retrieval. Ph.D. thesis, The University of Arizona (1995)
27. Minor, M., Tartakovski, A. and Bergmann, R.: Representation and Structure-Based Similar- ity Assessment for Agile Workflows, in Weber, R., O. and Richter, M., M.(Eds) CBR Re- search and Development, Proceedings of the 7th international conference on Case-Based Reasoning, ICCBR 2007, Belfast, NI, UK, August 2007, LNAI 4626, pp 224-238, Springer- Verlag, (2007)
28. Business Process Management Initiative (BPMI): BPMN 1.1: OMG Specification, Febru- ary, 2008, http://www.bpmn.org/, accessed October 2010.
29. OASIS: BPEL, The Web Services Business Process Execution Language Version 2.0. http://www.oasis-open.org/apps/org/workgroup/wsbpel, May 2006.
30. Workflow Management Coalition (WfMC): XPDL 2.1 Complete Specification (Updated Oct 10, 2008), http://www.wfmc.org/xpdl.html, accessed October 2010.
31. Kapetanakis, S., Petridis, Ma, J., Bacon, L.: Providing explanations for the intelligent mon- itoring of business workflows using case-based reasoning. In: Roth-Berghofer, T., Tintarev, N., Leake, D. B., Bahls, D. (eds.) Proceedings of the 5th International Workshop on expla- nation- Aware Computing Exact (ECAI 2010), Lisbon, Portugal (2010)

32. Kapetanakis, S.,Petridis, M.,Knight, B.,Ma, J.,Bacon,L. : A Case Based Reasoning Approach for the Monitoring of Business Workflows, 18th International Conference on Case-Based Reasoning, ICCBR 2010, Alessandria, Italy, LNAI (2010)
33. Kapetanakis, S., Petridis, M.: Evaluating a Case-Based Reasoning Architecture for the Intelligent Monitoring of Business Workflows, in Successful Case-based Reasoning Applications-2, S. Montani and L.C. Jain, Editors. 2014, Springer Berlin Heidelberg. p. 43-54.
34. Bandis, L., Kapetanakis, S., Petridis, M., Fish, A. (2017). An Architecture for Process Mining Using CBR on Rail Transport Industry. In proceedings of the 22nd UK CBR workshop, Peterhouse, December 2017,(Ed M. Petridis), Brighton press, pp. 11-18