Building an Integrated CBR-Big Data Oriented Architecture for Case-Based Reasoning Systems

Kareem Amin[1,2]

[1] PhD Candidate, German Research Center for Artificial Intelligence, Smart Data and Knowledge Services, Trippstadter Strae 122, 67663 Kaiserslautern, Germany,
[2] Big Data Consultant, Sulzer GmbH, Frankfurter Ring 162, 80807, Munich, Germany
kareem.amin@dfki.uni-kl.de,
kareem.amin@sulzer.de

1 Introduction

The growth of intensive data-driven decision-making is now being recognized broadly. Big data systems are mainstream and the demand for building systems that able to process data streams is growing. Yet many decision support systems act like ”black boxes”, providing little or no transparency in the rationale of their processes [1]. The ”black box” methodologies are not acceptable in crucial domains like health care, aviation, and maintenance. Experts prefer to reason the decisions. Current big data strategies tend to process in-motion data and offer many potential scenarios to work with. The big data term refers to dynamic, large, structured and unstructured volumes of data generated from different sources with different formats [2]. Therefore, it is a must for CBR systems that tends to process the in-motion data to manage their sub-tasks, such as collecting and formatting data, case base maintenance, cases retrieval, cases adaptation and retaining new cases [3]. In my research I will describe the idea of spanning the gap between CBR and Big Data based on the SEASALT architecture [4] [5]. SEASALT is an application independent architecture to work with heterogeneous data repositories and modularizing knowledge. It was proposed based on the CoMES approach to develop collaborative multi-expert systems and provides an application-independent architecture that features knowledge acquisition from a Web community, knowledge modularization, and agent-based knowledge maintenance. Its first research prototype was developed for the travel medicine application [4]. SEASALT aims to provide a coherent multi-agent CBR architecture that can define the outlines and interactions to develop multi-agent CBR systems.

2 CBR & Big Data

When CBR research has addressed increased data sizes, the primary focus has been compression of existing data rather than scale-up. Considerable CBR research has focused on the efficiency issues arising from case-base growth. As the case base grows, the swamping utility problem can adversely affect case retrieval times, degrading system performance [6]. CBR and Big Data collaboration is an emerging topic, some researches have been carried out focusing mainly on case base maintenance methods, aiming to reduce the case base size while preserving competence [8][7]. Few CBR projects have considered scales up to a million of cases [10][9]. The ability of case-based reasoning to reason from individual examples and its inertia-free learning makes it appear a natural approach to be applied to big-data problems such as predicting from very large example sets [6]. Likewise, if CBR systems had the capability to handle very large data sets, such a capability could facilitate CBR research on very large data sources already identified as interesting to CBR, such as cases harvested from the Experience Web [11], cases resulting from large-scale real-time capture of case data from instrumented systems [12], or cases arising from case capture in trace-based reasoning [13].

3 Research Focus

In my thesis I am going to concentrate on building a multi-agent CBR system that extends the SEASALT architecture. The proposed approach is designed to semi-automate the building of cases based on chunks of data coming from different streams, and being able to work with big number of historical cases stored in our case base. A real use case to elaborate the main goal of my model would be in manufacturing [14]. In manufacturing processes data comes from different machines and sensors. We need to detect any pattern that has led to a disqualified end product, and give a proactive solution to avoid or mitigate the effect of these kinds of patterns [15]. Hence, from the Big Data 4V's, I will mainly focus on velocity and volume with lower exposure to variety. I need to collect data from different sources and be able to detect patterns that match our old cases in real time. To achieve the aforementioned goals, the following objectives have to be fulfilled:

1. Extend the original SEASALT architecture with a new layer ”Knowledge Stream Management”
2. Correlate and synchronize between the chunks of data that come from different sources
3. Collect sufficient knowledge from domain experts that help in achieving point 2
4. Develop a methodology to apply the new approach to existing multi-agent systems as well as integrating it into the development of new multi-agent systems
5. Evaluate the new approach and the methodology within an industrial use case
6. Compare performance and accuracy with other existing techniques and systems

The proposed approach is roughly described in details in the following sections.

4 The Knowledge Stream Management

The original idea of the SEASALT architecture comes from Althoff, Bach and Reichle [5]. SEASALT consists mainly of five main layers, Knowledge Source, Knowledge Formalization, Knowledge Representation, Knowledge Provision, and Individualized Knowledge. Every layer contains several software agents designated for several tasks. Through my work, a new layer will be added: "Knowledge Stream Management" (See Figure 1). The new layer has two main tasks, the first is processing the streams of data coming from Knowledge Sources in real time, and the second is to give real time analysis to data patterns found within the streams. The Knowledge Stream Management layer will contain software agents designated for the prescribed tasks. System nodes [3] would be the available processing power. The Knowledge Provision layer will be distributed across several nodes, and hence each node contains Knowledge Provision agents. The Coordination Agent will act as the system manager who is aware of all the system nodes and responsible for the whole system control. He will be the data tap that uses the underlying framework to distribute the incoming requests across the system nodes. Normally, there are two kinds of nodes, one for Queries processing to retrieve results and the second for New Cases processing. It is possible to have up to N nodes in the system according to the volume of data that should be processed in real time. According to Big Data system architecture and sizing best practices provided from Hortonworks [4], for sustained throughput of 50MB/sec and thousands of events per second, we need 1-2 nodes and 8+ cores per node (more is better), 6+ disks per node (SSD or Spinning) and 2 GB of memory per node and 1GB bonded. In every node, there would be a Classification Agent to classify the received data chunks and assign it to the intended Topic Agent. Each Classification Agent is aware of the knowledge map gathered from knowledge sources and classify the incoming data according to predefined classes (collected before from domain experts). Then, the Classification Agent assigns the request to the intended Topic Agent(s). The Topic Agent is performing queries to retrieve the most similar cases. Since distributed nodes are being used in the hardware cluster, the Case Base will be replicated to avoid data integrity problems using replication channels to replicate data between all Case Base instances. The Case Factory agents will be centralized, and hence the Case Factory will have only one instance that performs case maintenance on a single Case Base. Afterwards, the results will be distributed to the whole system nodes using the replication channels.

We assume that solving big data problems will require also manual knowledge modelling. CBR - standing with one foot in the area of Machine Learning [automated knowledge generation] and with the other foot in the area of Knowledge-Based Systems [manual and semi-automatic knowledge modelling] -

[3] Every single node is a processing power

[4] Hortonworks is one of the biggest big data software companies founded in June 2011 as an independent company based in Santa Clara, California. http://www.hortonworks.com

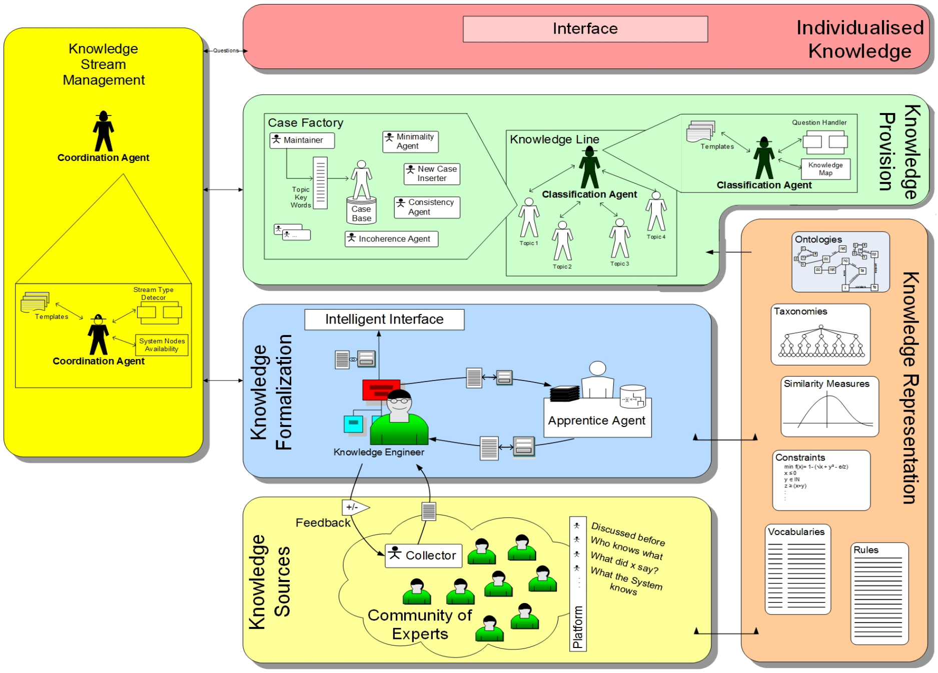

Fig. 1. Big Data Oriented SEASALT Architecture

is a natural candidate for finding a domain-and task-specific approach of integrating automated knowledge generation [using machine learning] with manual knowledge modeling [using knowledge-intensive CBR].

4.1 Potential Applications

1. Internet of Things (IoT) applications.
2. Ticketing systems and customer support applications.
3. Server logs anomaly detection applications.
4. Condition monitoring applications.

5 Current Progress & Future Directions

Currently I am shaping my PhD goals and approach. I intend to implement our approach and compare accuracy and speed performance with other case base maintenance methods. I am currently working to learn the big data system architectures and tools, that will help in the implementation phase. In the meanwhile, I am trying to find a suitable industrial use case to apply the proposed approach.

References

1. Herlocker, J.L., Konstan, J.A., Riedl, J.: Explaining Collaborative Filtering Recommendations. In: Proceedings of the 2000 ACM Conference on Computer supported cooperative work, ACM (2000).
2. A community white paper developed by leading researchers across the United States, "Challenges and Opportunities with Big Data," Purdue University, USA, 2012.
3. Aitor Mata, "A Survey of Distributed and Data Intensive CBR Systems," SpringerVerlag Berlin Heidelberg, pp. 582-586, 2009.
4. Meike Reichle, Kerstin Bach and Klaus-Dieter. Althoff, "Knowledge engineering within the application-independent architecture SEASALT" International Journal Knowledge Engineering and Data Mining, vol. 1.1, no. 3, pp. 202-215, 2011.
5. Kerstin Bach,Meike Reichle and Klaus-Dieter. Althoff, "A Domain Independent System Architecture for Sharing Experience," Proceedings of LWA 2007, Workshop Wissens- und Erfahrungsmanagement, September 2007, pp. 296-303
6. Vahid Jalali and David Leake, "CBR Meets Big Data: A Case Study of Large-Scale Adaptation Rule Generation," Case-Based Reasoning Research and Development, pp. 181-196, 2015.
7. Smyth, B., Keane, M.: Remembering to forget: A competence-preserving case deletion policy for case-based reasoning systems. In: Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, San Mateo, Morgan Kaufmann (1995) 377382
8. Smyth, B., McKenna, E.: Building compact competent case-bases. In: Proceedings of the Third International Conference on Case-Based Reasoning, Berlin, Springer Verlag (1999),329342
9. Daengdej, J., Lukose, D., Tsui, E., Beinat, P., Prophet, L.: Dynamically creating indices for two million cases: A real world problem. In: Advances in Case-Based Reasoning, Berlin, Springer (1996) 105119
10. Beaver, I., Dumoulin, J.: Applying mapreduce to learning user preferences in near realtime. In: Case-Based Reasoning Research and Development, ICCBR 2014, Berlin, Springer (2014) 1528
11. Plaza, E.: Semantics and experience in the future web. In: Proceedings of the Ninth European Conference on Case-Based Reasoning, Springer (2008) 4458
12. Ontanon, S., Lee, Y.C., Snodgrass, S., Bonfiglio, D., Winston, F., McDonald, C., Gonzalez, A.: Case-based prediction of teen driver behavior and skill. In: Case-Based Reasoning Research and Development, ICCBR 2014, Berlin, Springer (2014) 375389
13. Cordier, A., Lefevre, M., Champin, P.A., Georgeon, O., Mille, A.: Trace-based reasoning modeling interaction traces for reasoning on experiences. In: Proceedings of the 2014 Florida AI Research Symposium, AAAI Press (2014) 363368
14. S. Windmann, A. Maier, O. Niggemann, C. Frey, A. Bernardi, Ying Gu, H. Pfrommer, T. Steckel, M. Kruger and R. Kraus, "Big Data Analysis of Manufacturing Processes," in 12th European Workshop on Advanced Control and Diagnosis, 2015.
15. Mller, G., Bergmann, R.: Workflow Streams: A Means for Compositional Adaptation in Process-Oriented CBR. In: Proceedings of ICCBR 2014. Cork, Ireland, 2014