Knowledge based Business Contract Monitoring facility on Software Engineering

[1,2]Mohammed Saeb Khazaal, [1+]Chengliang Wang, [1]Ziaur Rahman

Abstract— Crowdsourcing has entered software engineering. In-house development, contracting, and outsourcing still dominate, but many development projects use crowd sourcing for example, to squash bugs, test software, or gather alternative UI designs. A Business Contract Monitoring is consisting of crowdsourcing. Their approach involves the Business Contract Monitoring (BCM) as a way to monitor contracts. Their focus is on the technical aspects of representing and viewing contracts. However, since BCM is includes the notions of events and temporal constraints, one can conceivably use an approach such as ours to help create a BCM specification based on a contract describing a service engagement. We develop a novel approach to work a hybrid of surface patterns, parsing, and classification to take the business events and their temporal constraints from contract text and then used, topic modeling for automatically organize the event terms into clusters.

Keywords— Crowdsourcing, Business contract monitoring, Contract Architecture, Software Development, Modeling

1 INTRODUCTION

The Software engineers goals is to use software development models for building software that meets user requirements and is delivered within the specified time limit and budget. Crowdsourcing is appearring form of outsourcing software development. Crowdsourcing is a name given to a revolution that marks the rise of online community composed of likeminded enthusiasts who work together, creating innovative solutions and lowering the production cost. The Crowdsourcing is to take the services of voluntary online community to build software rather than taking the services of traditionally employed workers. The main goal of software crowdsourcing is to produce high quality and low cost software products by harnessing the power of crowd. To view this objective, the crowd workers who accept to work on the task are given some financial or social incentives. The mission could be executed in a collaborative or competitive manner based on the organization style. Wikipedia and Linux are the collaborative crowdsourcing examples. Developing software through crowdsourcing blurs the distinction between a user and developer and follows a co creation principle.

In Business events indicate the essential processes involved in a service engagement as well as the risks and exceptions to consider. Moreover, the events are naturally temporally constrained; represent the conditions on their occurrence. The violation of a temporal constraint is often an important factor in contractual breach and the resulting complications.

To identifying and understanding business events and their temporal relationships in a service contract can help a business partner in successfully enacting a contract: that is, determining both what to other and what to expect from others. Understanding business events and their relationships can also potentially help it decide whether to enter into a contract in the first place. Note that real-life service contract are complex interactions with many nuances: we do not claim to have addressed all of the nuances just by identifying events and constraints from contracts, though what we do identify provide a necessary things for more elaborate future analyses.

Advantage:

- A good working relationship is maintained with the employee and Business owner
- Resource the job to meet all the commitment under the business contract
- Manage the contract good and consistently
- Be wary of perfectionism
- The standard you set is the standard you get
- Consciously develop relationships and trust

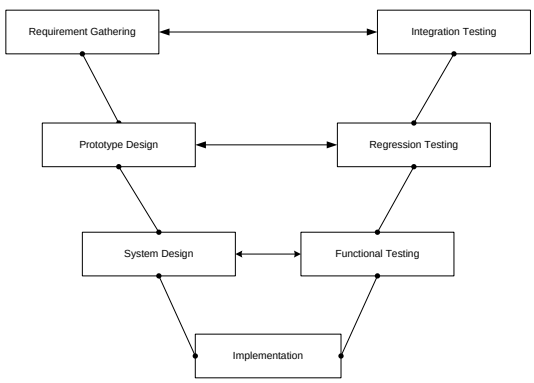

Figure 1: Software Engineering Process

- [1,2]*Mohammed Saeb Khazaal is currently pursuing master degree program in school of software engineering in [1]Chongqing University, China, PH-86-157300174141. College of Science, [2]Al-nahrain University, Iraq. E-mail: oody_1979@yahoo.com*
- [1+]*Wang Chengliang is currently professor in school of software engineering in Chongqing University, China, PH 86-23-65111964, E-mail: wcl@cqu.edu.cn*
- [1]*Ziaur Rahman is currently pursuing master degree program in school of software engineering in Chongqing University, China, PH-86-13042313140, E-mail: ziaurrahman167@yahoo.com*

1.1. MAIN ARCHITECTURE

In the flow of our approach, a hybrid of surface patterns, linguistic parsing and machine learning techniques. Contract Miner, first, takes raw online contracts as input, removes problem such as HTML tags and segments the contracts into collections. Second, it filters out sentences such as definitions and postal addresses that obviously do not contain business events and business monitoring. Third, it parses and prunes the remaining sentences to generate candidate events and constraints. Fourth, it applies machine learning on local and contextual features to individually identify true events and constraints from the candidates. Fifth, it applies topic modeling to extract hidden event topics.

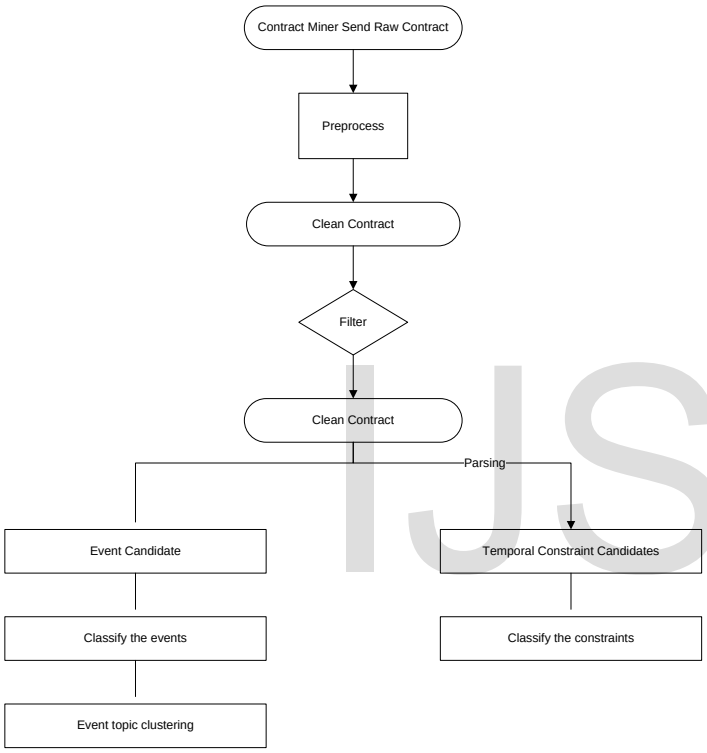

Figure 2: Main Architecture

2. ALGORITHM

The kNNq() algorithm is similar to Rq(), but it requires a dynamic radius lm to perform the pruning. In the beginning of the process, this radius is set to a value that covers all the indexed objects (line 1). It is adjusted when the answer set is first filled with m objects, or when the answer set is changed thereafter (line 12). Another difference is that there is a priority queue is just to hold the node not checked the subtrees from the nodes. All details are checked processing the single objects first (line 4 to 12) and then the subtrees (line 13 to 18).Include the subtrees, those closer to the query object that intersect the Query region are checked first (line 3). When an object closer than the m already found is located (line 8), it exchange the previous farthest one (line 11) and the dynamic radius is adjusted (diminished) to tight further pruning (line 12).

2.1. Algorithmic step

Require: root node Proot, the query object pq and number of objects m.
Ensure : Answer set with all objects satisfying the query conditions.

```
1.lm=∞
2.PriorityQueue.Add(Proot,0)
3.While((N=PriorityQueue.First())<=lm) do
4.      for each pj € N do
5.              if pj is a single object then
6.                      if | t (prep , pq) – t(prep , pj)| ≤ lm
then
7.                        calculate distance = t (pj , pq)
8.                              if distance ≤ lm then
9.                                 AnswerSet.Add (pj).
10.                                    if Answer-
Set.Elements() ≥ m then
11.                                AnswerSet.Cut (m).
12.                                    lm = Answer-
Set.MaxDistance().
13.                                end if
14.                           end if
15.                   end if
16.             end if
17.     end for
18.     for each pj € N do
19.             if pj is a substree then
20.                     if | t (prep , pq) – t(prep , pj)| ≤ lm
+ Rj then
21.                        calculate distance = t (pj , pq)
22.                             if distance ≤ lm + Rj then
23.                          PriorityQueue.Add( pj , dis-
tance).
24.                             end if
25.                     end if
26.             end if
27.     end for
28.end while
```

3. IMPLEMENTATION MODULES

3.1. BUSINESS EVENT EXTRACTION

A typical contract contains parts such as header, definition, body, and sign off. At the core of a contract are the clauses specifying mutual expectations expressed as normative relationships such as responsibility, powers, authorizations, prohibitions, and sanctions of the participating parties. Normative relationships express business relationships among the parties to service commitments and these normative relationships are built on top of business events. Our event extraction approach uses supervised classification. We select a set of 300 event candidates from the Noelle contract repository and manually label true business events. We emphasize that this repository contains genuine contracts that were entered into by real-life businesses. For privacy, some details, such as the amounts involved are redacted in this repository and replaced with? Characters—this deviation from the original contracts only

makes our task harder because such redactions cause parsing to become harder than it would be in actual contracts.

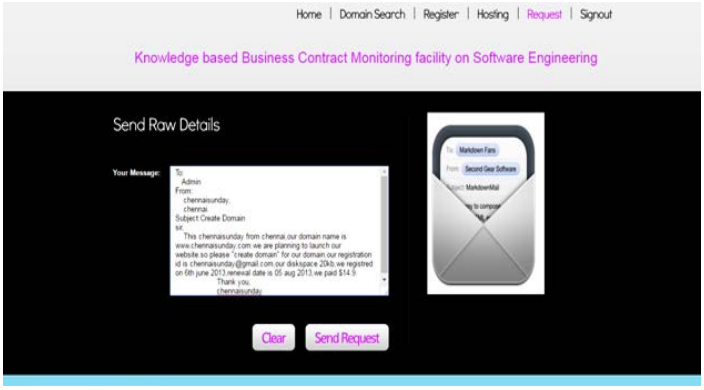

Figure 3: Business Event Extraction

3.2. Business Event Term Grouping

We evaluate Latent Dirichlet Allocation as applied in extracting business event clusters in two ways: centrality and clarity. First, we evaluate the ability of Latent Dirichlet Allocation to discover terms that are centered on a meaningful business event topic. We do so beginning with a human annotator assigning meaningful class labels to the automatically discovered terms groups. If the annotator is able to come up a descriptive label that covers the theme of a group of terms, it shows good centrality of the cluster. Second, we evaluate the separation of the terms clusters. We do so by using two independent human annotators matching a given list of class labels assigned by one of the authors to the term clusters. Terms of different themes should fall under different clusters. For our purpose of categorizing events by discovering event topics (or themes) and their corresponding descriptive vocabularies, we apply topic modeling in event categorization. In abstract terms, each event is regarded as a document; each document is a distribution of event topics; and each event topic is a distribution of event terms. Specifically, using the implementation of Latent Dirichlet Allocation, we extract prominent business event topics and representative vocabularies for each topic.

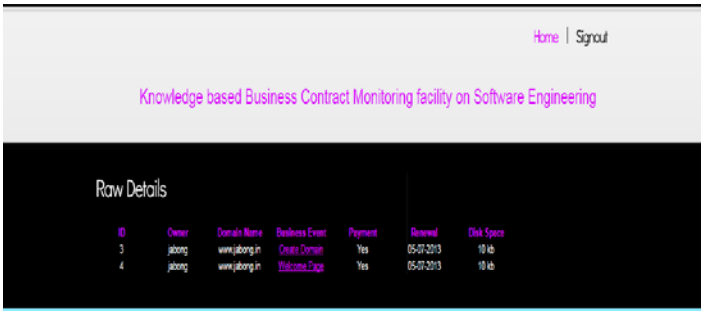

Figure 4: Business Event Term Grouping

3.3. Temporal Constraints Monitoring

Our evaluation demonstrates the effectiveness of machine learning methods for mining business events and temporal constraints. Supervised information extraction from service contracts faces unusual challenges. First, a contract is a legal artifact, and often exhibits more complicated nested structure and longer sentences than ordinary English text. Section and clause headings often cause the sentence boundary detector to break. The length of the sentences challenges the Stanford Parser to output the grammar tree. Second, an event is a subtle semantic unit that challenges automatic extraction. We define events as activities that capture essential business processes. Whereas other event extraction settings involve sentence selection, our events occur at the sub sentence level. Pruning helps reduce redundancy in a long legal sentence to capture the most important phrase that expresses an event. The extra processing enhances clarity but may lose information in some cases. Third, building a gold standard dataset is time consuming. Due to the lack of benchmark datasets relating to contracts, we built our own training corpus for event and temporal classification. Evaluation of the event topics is time consuming because there is no gold standard data available.

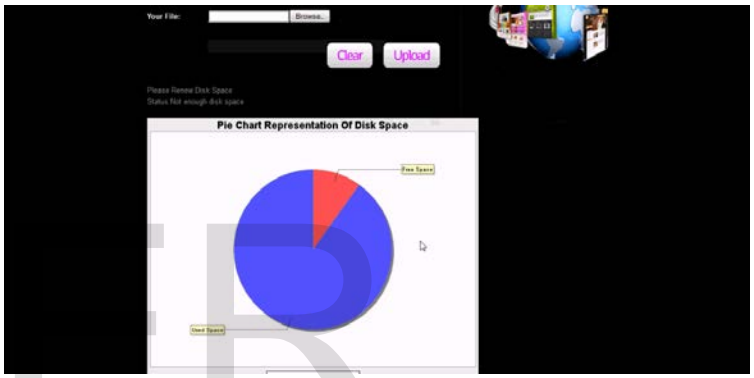

Figure 5a: Temporal constraints monitoring

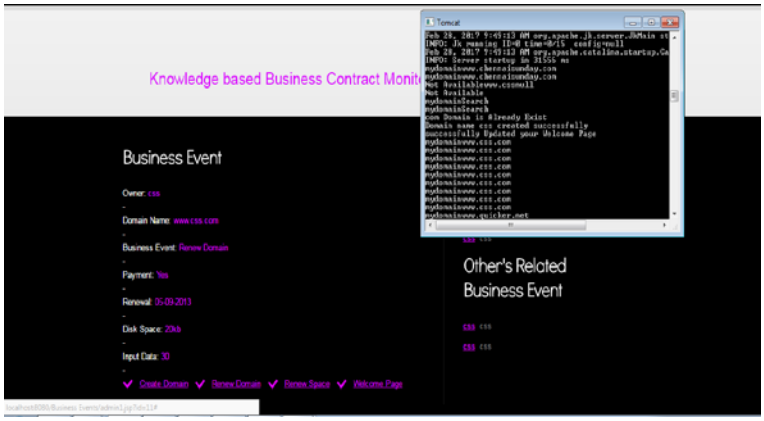

Figure 5b: Temporal constraints Monitoring

4. Code Implementation

4.1. Specify Business Event

```
try {
        Connection con1=databasecon.getconnection();
        PreparedStatement
ps=con1.prepareStatement("INSERT INTO rawcontracts
VALUES(?,?,?)");
        ps.setInt(1,id);
```

```
                ps.setString(2,rawdetails);
                ps.setString(3,status);
            int x=ps.executeUpdate();
            if(x!=0)
                {
                    re-
sponse.sendRedirect("ownerpage.jsp?message=success");
                }
                else
                {
                        re-
sponse.sendRedirect("ownerpage.jsp?message=fail");
                }
                }
                catch (Exception e)
                {
                        out.println(e.getMessage());
                }
```

4.2. Clustering

```
   try{
         con=databasecon.getconnection();
         st = con.createStatement();
        String qry ="select * from domainrequest where do-
mainname='"+url+"'";
        rs = st.executeQuery(qry);
        if(!rs.next()){
                out.println("Enter correct URL ");
        }
        else{
                exp=rs.getString("expdate");
                temp2=exp.split("-",3);

        if((Integer.parseInt(temp2[2]))>(Integer.parseInt(temp
1[2]))                          ||                          (Integ-
er.parseInt(temp2[2]))==(Integer.parseInt(temp1[2])))
                {

if((Integer.parseInt(temp2[1]))>(Integer.parseInt(temp1[1])) ||
(Integer.parseInt(temp2[1]))==(Integer.parseInt(temp1[1])))
                {

        if((Integer.parseInt(temp2[1]))>(Integer.parseInt(temp
1[1]))){
   welcomepage=rs.getString("welcomepage");
                re-
sponse.sendRedirect(site[2]+"/"+site[1]+"/"+welcomepage);
                }

if((Integer.parseInt(temp2[1]))==(Integer.parseInt(temp1[1])))
            {

if((Integer.parseInt(temp2[0]))>=(Integer.parseInt(temp1[0])) |
| (Integer.parseInt(temp2[0]))==(Integer.parseInt(temp1[0])))
            {
                welcomepage=rs.getString("welcomepage");

                re-
sponse.sendRedirect(site[2]+"/"+site[1]+"/"+welcomepage);
            }
            else{
                        //out.println(exp);
                                                re-
sponse.sendRedirect("home.html?Message=Site Expired");
                }
        }
                else{
                                                //out.println(exp);
                                                re-
    sponse.sendRedirect("home.html?Message=Site Expired");
                }
        }
                else{
                                                //out.println(exp);
                                re-
    sponse.sendRedirect("home.html?Message=Site Expired");
                                }
                }
                else{
                                //out.println(exp);
                                re-
    sponse.sendRedirect("home.html?Message=Site Expired");
                        }
        }
        con.close();
        st.close();
}
catch(Exception ex){
        out.println(ex);
}
```


5. Conclusion

We studied contracts as specifications of business monitoring. Business events and temporal constraints are resolvable to enacting a business monitoring, therefore extracting them is essential for each party to an monitoring to ensure it is being enacted correctly. Business events and constraints can be automatically analyzed to determine whether a potential service commitment is well-formed. Moreover, each party can check if the commitment is acceptable given its individual goals. Importantly, our techniques work on real-life contracts and can thus facilitate service commitments that arise in practice. Our classification-based extraction yields F-measures in the high 80% range and vocabulary clustering yields a 85% match with the gold standard.

6. Future Work

Business monitoring would be interesting to discover the dependency relationships across business events, e.g., if one event is a prerequisite of another. In the case of manufacturing, a down payment may be a necessary for product delivery and installment payments for continued product supply. Interlocked events form a network of business activities and lay the foundation for effective service commitment as a basis for

successful commerce. It is also worth studying the types of reliance on because these are associated with different (normative) business relationships. In these relationships can be categorized as normative relationships, such as commitments, permissions, and prohibitions. Events relate intimately to the antecedents and consequents in such relationships. Enriching the models in this manner can lead to improved requirements elicitation for service commitment as well as a principled basis for automating the service commitment life cycle from the perspective of a business partner.

References

[1] X. Gao, M. P. Singh, and P. Mehra, "Mining business contracts for service exceptions," IEEE Transactions on Services Computing, vol. 5, no. 3, pp. 333–344, Jul. 2012.

[2] H. Tanev, J. Piskorski, and M. Atkinson, "Real-time news event extraction for global crisis monitoring," in Proceedings of the 13th International Conference on Natural Language and Information Systems: Applications of Natural Language to Information Systems, ser. NLDB. London: Springer-Verlag, 2008, pp. 207– 218.

[3] M. P. Singh, "Norms as a basis for governing sociotechnical systems," ACM Transactions on Intelligent Systems and Technology (TIST), pp. 1–21, 2013, to appear; available at http: //www.csc.ncsu.edu/faculty/mpsingh/papers.

[4] H. H. Malik, V. S. Bhardwaj, and H. Fiorletta, "Accurate information extraction for quantitative financial events," in Proceedings of the 20th ACM International Conference on Information and Knowledge Management. Glasgow: ACM, 2011, pp. 2497–2500.

[5] M. Pasca, "Answering definition questions via temporallyanchored text snippets," in Proceedings of the 3rd International Joint Conference on Natural Language Processing, Hyderabad, January 2008, pp. 411–417.

[6] Leicht, N., Durward, D., Blohm, I., & Leimeister, J. M. (2015). Crowdsourcing in Software Development: A State-of-the-Art Analysis.

[7] Stol, K-J., and Brian Fitzgerald. "Research protocol for a case study of crowdsourcing software development." (2014).

[8] Milosevic, Zoran, et al. "On design and implementation of a contract monitoring facility." Electronic Contracting, 2004. Proceedings. First IEEE International Workshop on. IEEE, 2004.