Relations extraction on patterns lacking of Resulting Context

Asma HACHEMI[1], Mohamed AHMED-NACER[1]

[1]Computer science department, USTHB, Algiers, Algeria

ashachemi@usthb.dz, anacer@mail.cerist.dz

Abstract. Many software patterns are available nowadays. They allow the reuse of proved solutions in various areas of software engineering, but are expressed in different formalisms. This diversity is detrimental for patterns reuse, since it is difficult to compare and compose heterogeneous patterns (patterns expressed in different formalisms). Moreover, patterns composition is based on inter-patterns relationships, that are difficult to discern if they are not explicit. Thus, an automatic method that extract non explicit relations between patterns even if those latter are heterogeneous, becomes a necessity. In this context, we improve an existing method of automatic inter-patterns relations analysis. As many patterns lack of resulting context, our aim is to enable that method to extract relations on this kind of patterns.

Keywords : Patters formalisms, inter-patterns relationships, automatic relations extraction.

1 Introduction

Nowadays, the WWW supplies an increasing amount of knowledge covering diverse domains. Among others, it supplies a large number of software patterns that are a formidable tool allowing the reuse of proved solutions, in various areas of software engineering. A software pattern presents an issue to a recurrent problem, by offering a proven solution. Software patterns need to be composed, in order to solve complex problems that are not dealt by a single pattern. Inter-patterns relationships are on the basis of the patterns composition. However, it is difficult to discern these relations if they are not explicit in each pattern. Moreover, even if those relations are explicit, they are limited to intra-catalog relationships.

Indeed, a pattern is expressed through a pattern formalism, which is a syntactic structure of the pattern content. The majority of pattern formalisms in the literature differ in the number and degree of detail of their items. So, it is difficult to interpret and compare heterogeneous patterns. It is also difficult to compose them into larger solutions; a fact which is detrimental for patterns reuse.

This paper presents our approach that improves an existing method of automatic inter-patterns relations analysis. This method is the first automatic approach which handles relations between heterogeneous patterns, cross different catalogs. The aim of our improvement is to enable that method to handle more patterns formalisms; specially, those formalisms lacking of resulting context. So, works related to inter-patterns relationships extraction are described in section 2. Our approach for automatic relations analysis on patterns lacking of resulting context is presented in section 3. Finally, we conclude this paper and give some research perspectives in section 4.

2 Related works

Many researcher were interested in defining inter-patterns relationships, like [1], [2], [3], [4], … but very few works treat relationships extraction. In this area, Prabhakar *et al.* [15] propose a graphical model called Design Decision Topology Model, in order to represent design patterns and extract relationships between them; unfortunately, this work is limited to the analysis of relations on design patterns only. The method of Kubo et al. [10] is the first automatic approach able to extract relations on heterogeneous patterns, belonging to different catalogs, and is the only approach able to deal with different kinds of software patterns. Kubo *et al.* method is an interesting approach based on its own pattern model (consisting of *Starting Context, Forces, Resulting Context*), and on several text processing techniques (stop word removal [11], stemming [12], the TFIDF term weighting [13], vector space model [11] and the cosine similarity). However, Kubo et al. method is not able to treat patterns which lack of Resulting Context, so we propose to improve it to extract relations on this kind of patterns.

4 Our approach

Our approach towards an automatic way to analyze relations between patterns is based on Kubo et al. method, that we propose to improve in terms of the pattern forms handled. As many patterns do not express the Resulting Context explicitly (like those in [5], [7], [8], [9], [14], …), they cannot be represented and analyzed by the mo-del of Kubo *et al.*. Therefore, a value-added of our approach is the proposition of a solution to represent this kind of patterns and analyze relations on them.

Resulting Context is the result or product generated by the pattern application. So, each pattern has its resulting context either explicit in a dedicated section, or implicitly given in the Solution section. Here is an example of a resulting context expressed within the Solution : The pattern *Declare Before First Use* [8] aims to ensure that the declaration of an element is positioned before the reference to it (in an XML document). The resulting context of this pattern is the increase of the probability of treating the document in a single pass. Actually, this resulting context is expressed within the Solution section :*"This gives the processing software a better chance of doing a single pass traversal of the document".*

Our idea is to overcome the absence of a dedicated section for Resulting Context by using the Solution section, so that one be able to represent the third element of the pattern model (Resulting Context). A such use of the Solution does not alter the significance of the relationships that we are interested in (*Starting-Starting* [10], *Same*[10], *Resulting-Starting*[10], *Uses*[6] and *Refines*[6]). The reasons are :

- The relation *Starting-Starting* : The analysis of this relation is based on the Starting Context [10]. The Resulting Context is used only to represent the pattern in the Kubo *et al.* model. Thus, the use of the Solution section instead of the Resulting Context does not affect the meaning of this relation.

- The relations *Refines* : The analysis of this relation is based on the Starting Context and Forces [6]. The Resulting Context is used only to represent the pattern in the Kubo *et al.* model. So, the use of the Solution instead of the Resulting Context does not alter the meaning of this relationship.

- The relation *Same* : When two patterns share the same Starting Context and the same Solution, this means that these two patterns deal with the same problem and provide the same result. Thus, we can use the Solution section instead of the Resulting Context to analyze this relation.

  For example, let's consider the pattern Navigation Tabs [9] (called P1) which is Same as the pattern Navigation Tabs [7] (called P2). This relation is given by the author of [9], so we consider it as correct and process the analysis using our method. This latter starts by eliminating stop words [11] and applying the Stemmer [12] on the elements of P1 and P2. After that, the terms of these elements are weighted using the TFIDF method [13], and the cosine similarity is calculated as explained in [10]. Also, our method checks the inclusion [6], either it is true or false between each couple of elements. We obtain the results shown in Table 1.

Table 1. Results of comparisons between patterns elements

Compared Elements	Results
SC of P1 and SC of P2	Similarity = 0,934 SC of P1 includes SC of P2 = False SC of P2 includes SC of P1 = False
RC of P1 and RC of P2	Similarity = 0,956 RC of P1 includes RC of P2 = True RC of P2 includes RC of P1 = False
Forces of P1 and Forces of P2	Similarity = 1 Forces of P1 includes Forces of P2 = False Forces of P2 includes Forces of P1 = False
RC of P1 and SC of P2	Similarity = 0,136
RC of P2 and SC of P1	Similarity = 0,135

Since we obtain the similarity and inclusion results, we calculate the value of each relation between P1 and P2, using the definition of each relation (Uses [6] Refines [6], Same [10], Resulting-Starting [10], Starting-Starting [10]). For instance, the relation Starting-Starting between the patterns P1 and P2 is represented by the similarity value of the Staring Contexts of these patterns. The results of the relations analysis are shown in Table 2.

Table 2. Results of relations analysis

Relationship	Its value
P1 Uses P2	0
P1 Refines P2	0
P2 Uses P1	0
P2 Refines P1	0
Same	0,945

Starting-Starting	0,934
Resulting-Starting (P1 then P2)	0,136
Resulting-Starting (P2 then P1)	0,135

Finally, as in Kubo *et al.* method, the strongest relation of the eight types (P1 Uses P2, P1 Refines P2, P2 Uses P1, P2 Refines P1, Same, Resulting-Starting (P1 then P2), Resulting-Starting (P2 then P1), Starting-Starting) is assumed as the representative relationship. So we conclude in this case that the patterns P1 and P2 are *Same*.

- The relation *Resulting-Starting* : When the Solution of a pattern and the Starting Context of another are similar, this means that the second pattern (that we are interested in its Starting Context) can be applied after the first pattern (that we are interested in its Solution), because the solution of the first one provides the preconditions necessary to apply the second pattern. So, we can use the Solution instead of the Resulting Context to analyze the *Resulting-Starting* relation.

  For example, let's consider the patterns Titled Sections [14] (called P1) and Closable Panels [14] (called P2) related by the Resulting-Starting relation. This relation is given by the author of these patterns, so we consider it as correct and process the analysis using our method. We compare the elements of P1 and P2 and obtain the results shown in Table 3.

Table 3. Results of comparisons between patterns elements

Compared Elements	Results
SC of P1 and SC of P2	Similarity = 0,156
	SC of P1 includes SC of P2 = False
	SC of P2 includes SC of P1 = True
RC of P1 and RC of P2	Similarity = 0,119 RC of P1 includes RC of P2 = True RC of P2 includes RC of P1 = False
Forces of P1 and Forces of P2	Similarity = 0,097 Forces of P1 includes Forces of P2 = False Forces of P2 includes Forces of P1 = True
RC of P1 and SC of P2	Similarity = 0,248
RC of P2 and SC of P1	Similarity = 0,060

After that, we calculate the value of each relation between those patterns. The results are shown in Table 4.

Table 4. Results of relations analysis

Relationship	Its value
P1 Uses P2	0
P1 Refines P2	0
P2 Uses P1	0
P2 Refines P1	0,127
Same	0,137

Starting-Starting	0,156
Resulting-Starting (P1 then P2)	0,248
Resulting-Starting (P2 then P1)	0

Finally, considering the strongest relationship, we conclude that P1 and P2 are related via the relationship *Resulting-Starting*.

- The relation *Uses* : When the Starting Context and the Solution of a pattern are respectively included in the Starting Context and the Solution of another pattern, then this means that the second pattern *Uses* the first one. So, we can utilize the Solution instead of the Resulting Context to analyze this relation.

  For example, let's consider the pattern Extras On Demand [14] (called P1) which Uses the pattern Closable Panels [14] (called P2) according to the author of these patterns. So, we consider this relation as correct and process the analysis via our method. We compare the different elements of P1 and P2 and obtain the results shown in Table 5.

Table 5. Results of comparisons between patterns elements

Compared Elements	Results
SC of P1 and SC of P2	Similarity = 0,216 SC of P1 includes SC of P2 = True SC of P2 includes SC of P1 = False
RC of P1 and RC of P2	Similarity = 0,223 RC of P1 includes RC of P2 = True RC of P2 includes RC of P1 = False
Forces of P1 and Forces of P2	Similarity = 0,130 Forces of P1 includes Forces of P2 = True Forces of P2 includes Forces of P1 = False
RC of P1 and SC of P2	Similarity = 0,073
RC of P2 and SC of P1	Similarity = 0,114

After that, we calculate the value of each relation between those patterns. The results are shown in Table 6.

Table 6. Results of relations analysis

Relationship	Its value
P1 Uses P2	0,220
P1 Refines P2	0,173
P2 Uses P1	0
P2 Refines P1	0
Same	0,220
Starting-Starting	0,216
Resulting-Starting (P1 then P2)	0,073
Resulting-Starting (P2 then P1)	0,114

Finally, considering the strongest relationship, we conclude that the pattern P1 *Uses* the pattern P2.

5 Conclusion and perspectives

Our way of looking at the analysis of relations between patterns is based on Kubo *et al* method. We improved this method to enable it dealing with patterns which do not give their *Resulting Contexts* in an explicit manner. Our idea consisted of using the *Solution* section. As we have explained earlier, a such use does not alter the significance of the different relations treated. Some other improvements can be addressed to face the drawbacks inherent to Kubo *et al.* method, and to offer more benefits for patterns composition. Such as :

- The block HTML Analysis of the method is limited to the treatment of patterns expressed in HTML. This block can be extended to deal with patterns expressed in other ways.
- The method can be improved to treat patterns lacking of Starting Context and/or Forces, which are two necessary elements to represent patterns in the model of Kubo *et al.*
- The method can be extended to offer the functionality of Patterns Retrieval, which provides to a user having a particular problem, all available patterns that treat this problem.

References

1. Zimmer, W.: Relationships between design patterns. In: PLoP (1994)
2. Conte, A., Fredj, M., Giraudin, J.P., Rieu, D.: P-Sigma : un formalisme pour une représentation unifiée de patrons. In: Inforsid, Genève (2001)
3. Gnatz, M., Marschall, F., Popp, G., Rausch, A., Schwerin, W.: The Living Software Development Process. Journal Software Quality Professional, Volume 5, Issue 3 (2003)
4. Henney, K.: Patterns Inside Out. Talk presented at Application Development, London (1999)
5. Crumlish, C., Malone, E.: Designing social interfaces. available at http://www.designingsocialinterfaces.com/patterns/Main_Page (2009)
6. Hachemi, A., Ahmed-Nacer, M.: Primary inter-patterns relationships analysis. In: CARI2012, Algiers (2012) unpublished.
7. Yahoo design pattern library. http://developer.yahoo.com/ypatterns/.
8. Develop effective XML documents using structural design patterns. http://www.xmlpatterns.com/ (2000)
9. Lammi, J., Varjokallio, M., Hocksell, J.: A user interface design pattern library. http://www.patternry.com.
10. Washizaki, H., Kubo, A., Takasu, A., Fukazawa, Y.: Analyzing Relations among Software Patterns based on Document Similarity. In: IEEE Internationale Conference on Information Technology : Coding and Computing (2005)
11. Salton, G., McGill, M.: Introduction to Modern Information Retrival. MG-Hill Inc., NY (1983)
12. Paice, C.: Another stemmer. SIGIR Forum, Vol. 24, No. 3, pp. 56–61 (1990)
13. Salton, G., Yang, C.: On the specification of term values in automatic indexing. Journal of Documentation, Vol. 29, pp. 351–372 (1973)
14. Tidwell, J.: Designing interfaces - Patterns for effective interaction design. available at http://designinginterfaces.com/firstedition/ (2005)
15. Prabhakar, T.V., Kumar, K.: Design Decision Topology Model for Pattern Relationship Analysis. In: Asian PLOP 2010 Tokyo (2010)