A Model for the Management of the Knowledge applied to the Software Development Process

Vieira, Sandro C.
Banco do Brasil
Department of Information Technology
Ed. Sede IV – Brasilia / DF

Weigang, Li
University of Brasilia - UnB
Department of Computer Science
Darcy Ribeiro Campus

Ladeira, Marcelo
University of Brasilia - UnB
Department of Computer Science
Darcy Ribeiro Campus

Abstract
This paper describes a research which was conducted in order to obtain a model of the Text Mining and Case Based Reasoning designed for aiding the knowledge management for the software development process and improve software reuse. To achieve this goal we propose a model to determine the similarity between software development requests which are composed both by structured elements, described by codes or symbols, and non-structured elements described in natural language. An algorithm about the measurement of this similarity was proposed and the model was constructed as a complete information system. The application was realized as a case study in a finance organization. The results collected illustrate the efficiency of the system to improve software reuse and to avoid duplicated efforts in solution design. The paper shows the potential of the proposed system in the construction of detail solutions and characteristics for this organization.

Categories and Subject Descriptors
D.2.0 [**Software engineering**]: General

General Terms
Algorithms, Management, Measurement, Documentation.

Keywords
Text mining, Case based reasoning, knowledge management, software development process, similarity measurement.

1. Introduction

Software development, although not being the core for the majority of the enterprises, is so linked to business that a significant number of the organizations dedicate part of their resources to the activities related with the technology area. This situation is a result of the needing to develop products with an ever shorter life cycle, in addition to the requirement for strict controls aimed at maintaining the competitiveness before the rival companies.

In major companies, this reality has been experienced for many years, resulting that a considerable part of the applications currently in use originate from projects that were developed a long time ago. This situation is further aggravated by the lack of documentation pertinent to the applications in use, on account of the large number of interventions (maintenance) that are conducted in situations of emergency, and by the utilization of languages which do not provide the developer with the possibility of controlling and managing the evolution of the applications.

A significant part of the main systems utilized by these organizations suffers from this problem; they are usually oldies systems, with a high maintenance rate, inadequate or non-existent documentation, and the frequent need to interact with other systems of a different concept. A big part of these systems is written using structured language, and are executed everyday in mainframes installed in data processing centers of immense capacity. Their execution occurs in routines normally called Batch, on account of processing in lots. Also, it is common for these systems to possess interfaces based on the standard established by the 3270 terminals (with evolutions, such as the use of colors) but restricted to the presentation of texts with an area formed by 25 lines and 80 columns.

During the 90's, many computer science specilists said that the mainframes would be quickly replaced with smaller computers whose processing capacity has been continuously increasing since they were launched. However, if on the one hand the processing capacity has been increasing day by day, the same occurs with the demand for processing; the volume of transactions processed each day has increased in the same proportion, notably in companies of the financial branch – especially banks and credit card managers -, and, consequently the mainframes are still the choice of a great part of these institutions.

This phenomenon is justified by the many IT area specialists and executives' conviction that the performance of the this, executed in mainframes has not been caught up with in other platforms. Besides, the heavy investments that would have to be made to migrate the billions code lines currently estimated to be in production, and the risk associated with the replacement of

Manuscript received October 5, 2009
Manuscript revised October 20, 2009

applications that move billions of dolars in various data processing centers around the world.

Although a significant advance has been experienced in the area of software development, notably for the contributions of the software engineering, the management of the knowledge built in these systems still remains as a big problem to software developers. As an aditional chalenge, this knowlede managment should be done with a minimum efforts increase to the developers. This fact reinforces one of the concerns of this research, namely the adoption of mechanisms of artificial intelligence to minimize any work overload eventually incorporated in the process.

To adress this goals, a model is proposed to permeate the software development process, especially the business comprehension and requisite collection fases, in order to identify, with artificial intelligence aid, elements of previous solutions that may be reused.

2. Theoretical Foundation and Related Researches

2.1 Artificial intelligence

In his book Artificial Intelligence, Luger (2004) defines Artificial Intelligence (AI) as the capability of a machine to execute a certain task which, in order to be executed by a human being, would require the use of intelligence. This definition presents an interesting approach, as long as its comparison with an activity commonly present in the everyday life of people makes it easy to understand the expression, without the need for more complex definitions in technical terms, and is especially useful for the understanding of the objectives of the work, since the development of software is an intensely intellectual activity.

During this research, three lines of AI were investigated due to the capability of providing the necessary resources: case-Based Reasoning (CBR), neural networks and Bayesian networks. The preference was directed to the CBR line, due to its capability of providing in a simple manner a great part of the resources necessary for the identification of the similarities in the moulds of the proposal. Information Retrieval also was investigated due to its capability of simply determine text correlations.

2.2 Information Retrieval

This model, usually referred to in the literature as IR, was initially described by Salton (1971), and is presented as a simple and efficient alternative for the processing of elements described in natural language, which corresponds to the representation of the set of document terms in the form of a matrix A, with dimensions m by n (Am x n). Each line of the matrix corresponds to a document of the base, and each one of the matrix Ai,j entries corresponds to the relative frequency of each term of the document. According to Berry (2004), the greatest benefit of this model is that it is possible to explore the algebraic structure of the vectorial space; however, the dimensions m and n tend to grow rapidly, hindering the efficiency of the model. To solve this question, several alternatives have been proposed, ranging from the clustering of the terms in each document to probabilistic models, always aiming at the preservation of the information, as well as its representation in an easier and simpler model of treatment.

One of the possible approaches that is widely used in this area is the model known as TF-IDF (term frequency – inverse document frequency), which allows to process the elements present in a portion of the text, making the distinction of the relevance between the terms, considering the values of the frequency with which each term occurs on a base used for comparison. The central idea of this model consists in evaluating each one of the elements (terms) present in the fragment of the text being studied, considering, for that purpose, the frequency with which the term occurs in the text and compare to the frequency of this same term in the other cases recorded in the base. The more frequent a certain term is in the set of texts which compose the base of comparison, the smaller its relevance in the data mining. This means to say that a term that occurs few times in the base of cases and that occurs many times in the document will have, for purposes of comparison, a greater relevance than the terms occurring more frequently in the documents.

2.3 CBR and Textual-CBR

The case based line of reasoning presents an approach model of the way the human beings utilize their brains to infer results from their past experiences. Its proposition is attributed to the works initiated by Kolodner (1993), which formed the bases for this type of problem solving, and which glimpsed a new frontier in the field of artificial intelligence, and at the same time reduced noticeably the dependence of a specialist in the domain of application for the production of rules, normally utilized by specialist systems.

The work of Kolodner materialized the ideas previously defended by Schank (1991) which associated the process of learning to that of memorization, and described what she believed was one of the ways commonly used by the human brain. The cognitive theory prevailing at the time of her studies pointed in the direction of the symbolic learning and associated the intelligence to the capability of manipulating these symbols through the utilization of rules.

Derived from the line of researches in CBR, the area known as Textual-CBR, or just TCBR, is presented as an option for the treatment of situations in which the cases cannot be described in discreet or discreet-able terms. For that, approaches are proposed that aim at processing the textual elements and extract from them the items of information relevant for the composition and identification of cases. This line of research is presented as an alternative to the usual text processing mechanisms, which, as a rule, are based on vectorial-space method, initially proposed by Salton (1984), or on its variations. Lenz (1998) establishes an important comparison between the TCBR and the traditional IR mechanisms. The results have a special significance for this research, considering that the processing of textual information is one of the situations that will be dealt with. The work of Lenz was deepened by Brüninghaus (2001) who analyzed the role of the Information Extraction (IE) in the context of the TCBR application. In his approach, the important evolution of the Natural Language Processing (NLP) techniques was considered, as they are capable of generating vocabularies about specific domains from a small set of samples, according to the author.

Although the results obtained by Brüninghaus were considered as successful in his research, from the reading of the articles mentioned, it can be gathered that the amount of efforts necessary for the construction of the set of rules of the use of TCBR was not small, as well as the dependence of the rules in relation to the application domain is highly evidenced. This observation influenced on our decision to propose a hybrid model, in which the processing of textual elements will take place with the utilization of IR techniques based on the propositions of the vectorial-space model.

3. Proposed Model

Software development and maintenance is generally regulated by a formalism that is an indication of its need by an area of the organization. This formalism usually materializes by means of an IT service request, which synthesizes the main characteristics and functionalities to be presented by the solution. The objective of this model is to obtain an index, from the data available in the service request, capable of expressing the similarities between two or more requests: one of them at the moment of its elaboration and the other ones recovered from the case base. For that purpose, it is necessary to define a set of components which supports the retrieval of the information and the measuring of the similarities between the cases considered.

Figure 1 depicts, in general terms, the functioning of this model. The flow of activities starts with the generation of a new service request, which may be addressed to the building of a new application (or functionality) or to improve an existing functionality.

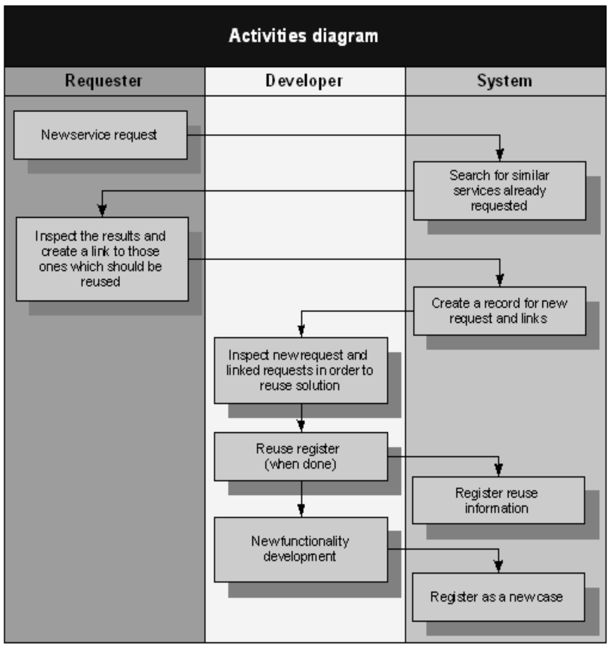

Activities diagram
Requester
Developer
System
Newservice request
Search for similar services already requested
Inspect the results and create a link to those ones which should be reused
Create a record for new request and links
Inspect new request and linked requests in order to reuse solution
Reuse register (when done)
Register reuse information
Newfunctionality development
Register as a new case

Figure 1 –Model activities flow

Starting from the new service request (being elaborated by the requester), the system must be capable of determining the similarity with the other cases recorded in the base. In order to make this determination, it is necessary to consider both the structured elements – which may be described in numerical or symbolical terms – and the non-structured, represented by texts blocks written in natural language. All the requests recorded in the base must be compared. in order to identify for those presenting the highest rate of similarity; after such comparison has been made, the similar cases have to be shown to the requester, who will be able to evaluate them and link to the request being elaborated the cases deemed relevant (which present similarity) for a possible re-utilization at the moment the request is accommodated.

Since the processing of the requests implies the evaluation of elements of a rather different nature – the structured ones are easily comparable, while the non-structured require a complex process -, a first project decision which was to segregate those elements and treat each grouping separately. That decision was made because, considering that the comparison of the structured elements is a much easier task than the comparison of the non-structured elements, it may function as a filter so that only the pre-selected cases be thoroughly evaluated, thus reducing the computational effort necessary for the computation of extensive bases.

3.1 Treatment of the structured elements

The task relative to these elements is to determine a rate of similarity between the new request, from now on called case study, and the other recorded requests, which will be referred according to the case base. The settlement of this rate must consider all the structured elements present in the request and take into account that each one of the elements may have a distinct contribution in the composition of the similarity rate. For that purpose, and meeting those requisites, the first element for the evaluation of the similarity rate between the requests was built, which will be defined here as similarity rate between the structured elements (ISee), whose calculation will be made through the formula indicated in (1), where the letter "A" is a reference of the new request and the letter "X" is used for the designation of a certain request retrieved from the case base.

$$ISee(A,X) = \sum_{i=1}^{i=num-elem} ISe_i(A,X) \;\; x \;\; W_i \qquad (1)$$

where : $ISee(A,X)$ indicates a similarity index of all the structured elements of A and X documents;
$ISe_i(A,X)$ indicates the similarity index of element i from case A when compared to the same (structured) element on document X ;
W_i is the weight associated to element i;
$num-elem$ is the total amount of elements to be compared;

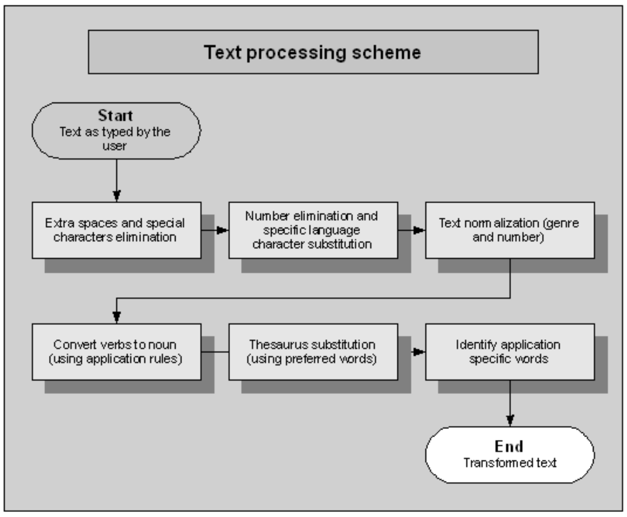
Formula 1– Compute of the similarity rate between structured elements.

For each pair of compared requests a first similarity rate will then be obtained, calculating the similarity of each one of the structured elements in the two compared requests, weighed according to a value. These relative weights of each one of the structured elements of the request will be initially estimated considering the opinion of specialists in the application domain, and will be refined during the phase of validation and tests. All the base cases will be compared with the case being studied for the determination of the rate, and only the ones that obtain values above a certain threshold will have their structured elements compared.

3.2 Treatment of the non-structured elements

The non-structured elements are an extremely important source in the evaluation of the final similarity between the requests, and their treatment must be such that allows considering the largest possible amount of relevant pieces of information of the objective and functionality fields of the IT service request.

These texts are pre-processed so as to eliminate elements which are non-significant for the determination of similarity. A further option was to make during this phase the enrichment of the text by means of the identification and clear definition of the relevant terms in the application domain. This set of terms was initially obtained with the aid of specialists and will be incremented by means of the utilization of techniques which allow identifying and separating them for classification. The scheme utilized for the processing of the text blocks is depicted in Figure 2. From the text resulting from this processing the matrices of frequencies of words (usually referred to in the literature as *bag of words*), composed of a term and a number representing the frequency of this term in the case being studied, and which represent the base on which the comparison of text elements will be made.



Text processing scheme
Start
Text as typed by the user
Extra spaces and special characters elimination
Number elimination and specific language character substitution
Text normalization (genre and number)
Convert verbs to noun (using application rules)
Thesaurus substitution (using preferred words)
Identify application specific words
End
Transformed text

Figure 2 – Scheme for the processing of text blocks

Once the terms and respective frequencies are obtained, and the frequency of all the terms which are already in the case base is known, we can determine the significance of each one of these terms, by applying the formula depicted in (2).

$$S_i(A) = tf_i(A) \;\; x \;\; IDF(T) \qquad (2)$$
$$S_i(X) = tf_i(X) \;\; x \;\; IDF(T)$$

where : $S_i(A)$ is the significance value of the term i in case A;
$S_i(X)$ is the significance value of the term i in case X (recovered from the base);
T is the term i from the bag of words;
$tf_i(A)$ is the term frequency in case A;
$tf_i(A)$ is the term frequency in case X;
$IDF(T)$ is the inversed frequency of term T in case base;

Formula 2 – Determination of the significance of the terms of the documents to be compared.

The calculation of the IDF value is made by means of well known elements of the information retrieval techniques, utilizing the concepts introduced by Salton (1984) in the proposition of the vectorial-space model.

$$IDF(T) = \ln \frac{\text{Qtde\_total\_dcto}}{\text{Qtde\_dcto} \subset \text{T}} \qquad (3)$$

where : $IDF(T)$ is the inversed frequency associated to term T;
Qtde_total_dcto is the total amount of documents in the case base;
Qtde_dcto $\subset$ T is the amount of documents where term T is present;

Formula 3 – IDF determination for each one of the terms of the base.

The application of this formalism in the model proposed comes from the definition of this value for each one of the terms existing in the documents recorded in the case base, something that is done with the utilization of the formula (3).

Upon defining the significance of eachterm in each one of the documents being compared, the measurement of the similarity between them is made by means of the formula (4), which identifies the distance between the relative significances.

$$ISn_i(A,X) = S_i(A) \;\; x \;\; \sqrt{1-(S_i(A)-S_i(X))^2} \qquad (4)$$

where : $ISn_i(A,X)$ indicates the similarity index of element i from case A when compared to the same element on document X;
$S_i(A)$ is the significance value of the term i in case A;
$S_i(X)$ is the significance value of the term i in case X (recovered from the base);

Formula 4 – Determination of the similarity rate between the terms of two documents.

The final stage of the evaluation of the similarity is to make a summation of the relative similarity rates of each one of the terms of the list, obtaining the global similarity rate for the non-structured elements. The formula regarding this operation is described in (5):

$$ISne(A,X) = \sum_{i=1}^{i=num-term} ISn_i(A,X) \qquad (5)$$

where : $ISne(A,X)$ is the similarity index of all non-structured elements;
$ISn_i(A,X)$ is the similarity index of non-structured term i in document A when compared to the same term of document X;
$num-term$ is the total amount of terms compared in the bag of words;

Formula 5 – Ssimilarity rate between non-structured elements.

The ISne rate represents the existing similarity between the non-structured elements in the two compared cases (referred to in the formulae as A and X). The composition of this rate with the one obtained for the structured elements (ISee) is this made for the obtainment of the final similarity rate between the two compared requests.

The composition between the rates must take into account the existence of distinct weights for the structured and non-structured elements. Considering all these elements, we, then, have the final similarity formula between the requests, which is indicated in (6).

$$IS(A,X) = \frac{ISee(A,X) \;\; x \;\; Wee \;\; + \;\; ISne(A,X) \;\; x \;\; Wne}{Wee \;\; + \;\; Wne} \qquad (6)$$

where : $IS(A,X)$ is the final similarity index of document A when compared to document X;
$ISee(A,X)$ is the similarity index of all structured documents;
Wee is the relative weigth of structured elements;
$ISne(A,X)$ is the similarity index of all non-structured documents;
Wne is the relative weigth of non-structured elements;

Formula 6 – Final similarity rate between two documents.

The stage of identification of similar cases is closed when the requests were compared (considering the existence of the threshold for comparison of non-structured elements) and a list is made, sorted in descending order, of the similar requests for presentation to the requester, who will then browse through the various requests and link the requests chosen by him to the new one that is being elaborated.

4. Implementation

The proposed model establishes a set of procedures and methods aimed at finding out the similarity between two service requests. For its evaluation and support, a tool has been developed which is called FSSAIA (an acronym in Portuguese for “tool for the request of services assisted by artificial intelligence”). In the development of this tool, the utilization of free software solutions and the its utilization in the Web environment were prioritized as a way of easily integrating it with other tools for the control of service requests.

The requester interface is the first element of interaction with the aplication tool, and the starting point for the whole process. It is responsible for the initial record of the request which will be processed and for interacting with the support mechanisms for the obtainment of similar cases. Among its attributions there is the one of implementing a basic form for the recording of the request, with the structured and non-structured fields identified in the modeling phase and the functions necessary for the recording of this new request in the system database.

Support to the model proposed is provided by a set of components which have the function of processing the service request under elaboration and transforming it in a case capable of being treated by the application. This work starts in the pre-processing of the request, which makes the segregation of the structured and non-structured and activates the text treatment module for the processing of the non-structured elements.

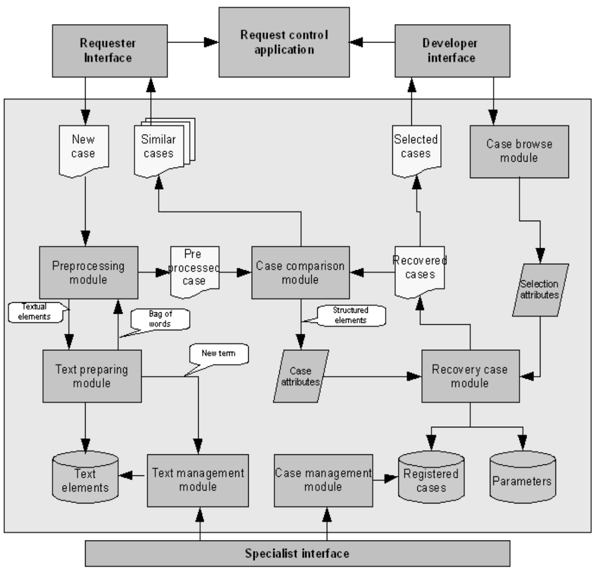


Figure 4 – Model support application architecture

The text treatment module applies the process described in section 3.2 by converting the texts block in a bag of words and respective frequencies. During this processing, an enrichment process is aplied to the text, with the recognition and replacement of terms common to the application domain, the removal of terms of little significance (stop words). The terms which are in the text block to be processed, and that are not part of the base of text elements, are inserted in the base with the marking that they have to be classified by the specialist. This marking is part of the incremental process of learning and construction of a base of terms which is adequate for the application domain. At the end of this processing, the word matrix is returned to the activator module, which, by associating the matrix with the structured elements makes the composition of the new case (pre-processed case) for comparison with the other ones recorded in the application base.

The case comparison module is then activated to determine the similarity rate between this new case and the other cases in the base. This comparison is initially made with the comparison of the structured elements and with the assembly of a case list whose similarity rate between these elements is above a certain threshold; only the cases in this situation will have the non-structured elements compared for the composition of the final similarity rate. This decision was made to optimize the process, since the comparison of non-structured requeires a lot of processing time. After the comparison is done, an ordered list of similar cases will be displayed to the demander for verification and possible linking to the request that is being elaborated.

At this moment, the demander has to evaluate, among the cases identified by the application, which ones may present significant contributions or relevant information which may facilitate the accommodating of his/her request; the cases that fit this situation should be marked by him/her as relevant and linked to the new request.

After the submission of a new service request is completed, the cycle continues with the administrative definition of its priority to be accommodated. This stage involves a negotiation between the demander and service units and it is not represented in the process.

When the request starts being accommodated, a manager is defined who is responsible for its monitoring. A team is also assigned to assist him. The size of the team depends on the complexity of the request, the deadline to accommodate the request and the relevance of the subject for the organization. This manager must evaluate the aspects involved in the accommodating of the request, elaborate schedules and define the necessary resources. At this point, the evaluation of similar requests may earn an immense gain for the development process, whether on account of the possibility of re-utilization of solutions already developed, or on account of the information about legal aspects which must be considered in accommodating the request. The access to these pieces of information occurs by means of the consultation of the linked requests and may be complemented by researches made by the executer in the base of recorded cases.

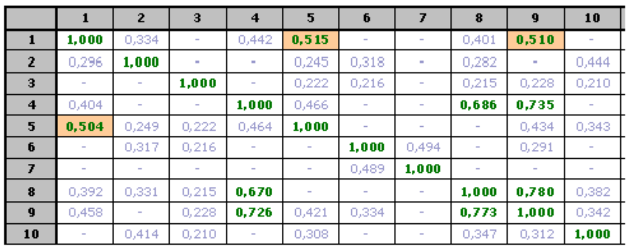
The interface with the specialist provides the necessary results for the management of the bases of text elements and of cases. The new terms identified by the text treatment module are inserted with the information that they need to classified, and this procedure may add a significant value to the recognition of similarities. A text block containing the term “credit card” at a first processing would generate a list with those two words treated in a distinct manner; if the application domain is the financial environment, as in the example treated in the experiment that was made, it is convenient that the tool be apt to recognize those words as a term of the application domain. Also, the case base has to be managed, so as to prevent its disorderly growth; the maintenance of very old cases or with an irrelevant contribution needs to be periodically evaluated as a way of preserving the application capability of providing adequate responses in a reasonable time.

5. Experiments and Results

In order to evaluate this model an experiment was proposed, and conducted with a base of documents composed of 50 requests obtained from actual cases that was imported for the tool described in the previous item. This experiment was divided into two stages which aimed at adjusting the model to the selected domain and evaluating its capability of identifying similarities between those requests and new ones.

The first stage of the tests was conducted in accordance with the CRISP-DM (2008) reference model and had the objective of obtaining a similarity matrix resulting from the comparison among all the requests. The data (here represented by the requests already accommodated) were collected form the records maintained by the target organization of the experiments in a system that controls the progress of the requests. It was chosen a group of requests destined to one of the request fulfillment managements of the organization whose characteristics are similar to the other managements, so that the results obtained can be generalized.

The comparison produced a similarity matrix of 50 x 50 dimensions; to illustrate the experiment and allow the understanding of this stage, the figure 5 depicts a small part of this matrix, whose results are explained below.



	1	2	3	4	5	6	7	8	9	10
1	**1,000**	0,334	-	0,442	**0,515**	-	-	0,401	**0,510**	-
2	0,296	**1,000**	-	-	0,245	0,318	-	0,282	-	0,444
3	-	-	**1,000**	-	0,222	0,216	-	0,215	0,228	0,210
4	0,404	-	-	**1,000**	0,466	-	-	**0,686**	**0,735**	-
5	**0,504**	0,249	0,222	0,464	**1,000**	-	-	-	0,434	0,343
6	-	0,317	0,216	-	-	**1,000**	0,494	-	0,291	-
7	-	-	-	-	-	0,489	**1,000**	-	-	-
8	0,392	0,331	0,215	**0,670**	-	-	-	**1,000**	**0,780**	0,382
9	0,458	-	0,228	**0,726**	0,421	0,334	-	**0,773**	**1,000**	0,342
10	-	0,414	0,210	-	0,308	-	-	0,347	0,312	**1,000**

Figure 5 – Part of the similarity matriz obtained

Each line of the matrix presents the result of the comparison between the request indicated and all the other requests; the main diagonal of this matrix shows the result of the comparison of the request with itself and shall present, therefore, the unitary value meaning that the objects of the comparison are identical. The cells marked in bold type indicate values above 0.5 (five-tenths) – considered similar according to the convention established. The cells in which these values appear crosshatched indicate that there was difference between the indication of the system and of the specialists that evaluated the requests. Values above 0.7 (seven-tenths) indicate that the requests compared present a high similarity, also in accordance with the convention established by the specialists in the application domain.

The values used in this comparison are indicated in table 1 and were obtained by the agreed indication of the specialists involved in the process of evaluation of the application.

Indication of the specialist	**Range of values**
Low or no similarity	0.00 to 0.49
Medium similarity	0.50 to 0.69
High similarity	0.70 to 1.00

Table 1 – Range of values for comparison of results of the system

For the evaluation of the results in this stage, a confusion matrix was used, built from the comparison between the expected and obtained results; this comparison aimed at determining the degree of accuracy of the system in relation to the previous indication available. The results were computed in each one of the cells, in accordance with the classification of the system and the previous classification made by a specialist.

Expected (specialist) \Obtained (system)	**Positive**	**Negative**
Positive	TP	FN
Negative	FP	TN

Table 2– General formation of the matrix utilized

In the evaluation of the results, the confusion matrix was organized in the following manner:

- Lines: each line expresses the result of an expected classification, according to a previous evaluation made by specialists in the problem domain;
- Columns: each column presents the results obtained by the system, with the application of the model for the expected classification;

TP: represent the true positive values, that is, those values in which the classification automatically obtained by the system coincided with the expected value for the elemnts pertaining to the class being treated;

TN: represent the true negative values, that is, the values in which the expectation based on the indication of specialists was their non-belonging to the treated class and thus they were correctly classified;

FP: represent the false positive values in which the classification made by the system indicated that they belonged to the treated class while the specialists expected otherwise;

FN: represent the elements classified by the system as not belonging to the class, while the specialists classified them as belonging to the class.

The classification made by the application was effected as shown in table 1, which presents three distinct classes: non-similar, medium similarity and high similarity. For the obtainment of the rates representing the degree of accuracy of the system a similarity matrix was built for each class being studied, and, from these results, a resulting confusion matrix was elaborated, as indicated in Table 3.

Expected \ Obtained	**Positive**	**Negative**
Positive	806 (0.329)	10 (0.004)
Negative	12 (0.005)	1622 (0.662)

Table 3 – resulting confusion matrix

This stage allowed us to make the necessary adjustments and refinements in the relative weights of each one of the attributes considered and ended with the obtainment of accuracy, sensitivity and specificity rates in the order of 99.1%, 98.8% e 99.2%, respectively, obtained as described below:

a) Accuracy (A) = 100 * (TP+TN) / (TP+FP+TN+FN)

b) Sensitivity (S) = 100 * (TP)/(TP+FN)

c) Specificity (E) = 100 * (TN)/(TN + FP)

Although it is important to point out that these results have been influenced by the prevalence of documents in the non-similar class, among the documents that were considered similar, the rates were also above 90%.In the second stage of the test, some new requests were selected to be processed by the tool of support to the model, divided into three scenarios, according to the previous classification of the similarity rates presented. In this stage, a set of requests identified as belonging to each one of the scenarios selected (scenario 1: low or no similarity; 2: medium similarity; and 3: high similarity) was processed for a detailed analysis of the results obtained and for the establishment of a standard of behavior relative to each one of these cases. The results obtained for each one of these scenarios are briefly indicated.

The first scenario evaluated was composed of requests that did not present similar cases in the case base maintained by the system, according to previous evaluation made by specialists in the application domain. Two cases were evaluated in this scenario: the results obtained by the application for the first case confirmed the inexistence of similar requests; as for the second case relative to this scenario, it had the indication of a similar case and, therefore, we dedicate additional attention to the study of this result. In the second comparison made for testing, there was the occurrence of a case identified as similar by the system in opposition to the expectation that all the comparisons would result in rates below 0.499. In the evaluation of this result, a few details clarify the situation: first, it is necessary to consider the quality of the specification contained in the request, which presented a rather poor description of the functionalities to be implemented. This situation hinders the capability of the model in the identification of similarities due to the existence of just a few terms to be compared; when a request in this situation is compared with other ones containing lots of details, and consequently with lots of terms common to the domain, it is usual that these terms raise the similarity rate between the non-structured elements and, as a result of this raising, the final rate is also increased. Another consideration to be made regarding this comparison is that, although it was considered non-similar by the specialists, the new requested deals with the provision of information to an external organization; although dealing with other items of information, obtained from another system and destined to another entity, the suggested case also deals with the provision of information. Therefore, there is similarity between the functionalities requested, although it is estimated that it would be of no use for the executer of the service to take advantage of any item of information present in that case. In this situation, once identified the low possibility of re-utilization, the very demander could make the option of not linking the requests.

The second scenario was composed of cases that glimpsed similarities between the cases present in the base and the cases to be evaluated. The expectation for these cases was that the application would be able to identify those similarities, something that really occurred. In the case taken as an example, the result obtained by the application was totally consistent with the indication of the specialists, resulting in the identification of two similar cases which, according to this indication, could be utilized to support the process of fulfillment of the request identified in the new request.

In the scenario of the requests with high similarity, it was expected that the request would find in the system case base some request for which the similarity rate would lie somewhere above 0.700 according to the parameters set for the final similarity rate. The existence of similarity between the evaluated request and four other requests present in the case base, with the case in question having high similarity. This result was considered excellent by the specialists that evaluated it, as the suggested request was rather similar to the new request and required similar information, obtained from the same system but destined to different sections within the organization. The re-utilization of functionalities in this case would be practically certain and would imply a considerable economy of efforts. In the processing of this request, a highlight was given to the identification of two other requests which presented similarities in relation to the current request, which could also help in the development of the solution. Another request in the case base, although not considered highly similar in the evaluation of the specialists, presents a high degree of similarity between the structured elements and describes functionalities also inherent to the system being treated. It has to be considered that the obtained rate for this case was only 0.501, something that places it in a region very close to the defined boundary.

6. Conclusions

This research involved the identification and characterization of a problem frequently experienced by the IT professionals and organizations witch consists in the lack of management of the knowledge linked to the software development process. For its solution, a model was proposed with specifically planned characteristics to allow the identification and re-utilization of similar already developed solutions.

Although applied on a relatively small case base, it was possible to identify similarities that, in principle, would not be perceived unless the team responsible for the development of the new solution had also participated in the previous solution. As software development teams composition is subject to frequently changes, an interstice of one or two years would mean that the entire team have been replaced in their functions. Increasing demand for software products compel the IT managers to improve their available resources utilization, so as to be able to fulfill the prioritized demands of the organizations.

It must also be considered the increasing enterprises adoption of project management practices in the conduction of software development teams, with the adoption of matrix teams of specialists in each one of the activities (requirements specification, data modeling, coding, etc.) to the detriment of the maintenance of fixed teams focused on a certain product. This situation, while on the one hand allows to explore efficiently the skill of each professional in the group, directing him to the set of activities in which he may have developed some type of specialization, be it through either a theoretical or practice training, on the other hand requires the adoption of efficient mechanisms for the management of the knowledge involved in the process under penalty of hindering future maintenances or even the continuation of business, under certain conditions.

The results obtained through the application of the proposed model demonstrated the existing potential for its utilization and are rather promising in relation to the benefits it may bring for the software development.

References

[1] BERRY, Michael W. Survey of text mining: clustering, classification and retrieval. 1st edition, New York, USA. Springer, 2004.

[2] BRÜNINGHAUS, Stefanie; ASHLEY, Kevin D. The role of information extraction for textual CBR. Proceedings of the 4th International Conference on Case-Based Reasoning. Vancouver, BC, Canada , 2001.

[3] CRISP-DM. Cross Industry Standard Process for Data Mining. Disponível em: http://www.crisp-dm.org. Acesso em 22 de junho de 2008.

[4] LUGER, G. F. Inteligência Artificial: Estruturas e estratégias para a solução de problemas complexos. 3ª edição. Porto Alegre, Brasil. Bookmann, 2004.

[5] KOLODNER, Janet L. Case-Based Learning. 1st ed., Los Altos, USA. Morgan Kaufmann, 1993.

[6] LEAKE, David B. Case-Based Reasoning: Experiences, Lessons, and Future Directions. 1st edition, The MIT Press, 1996.

[7] LENZ, Mario. Textual CBR and Information Retrieval – A comparison. Proceedings of 6th German Workshop on CBR, 1998

[8] PRESSMAN, Roger S. Engenharia de Software. 6ª edição. São Paulo, Brasil. McGraw-Hill do Brasil, 2006.

[9] SOMERVILLE, Ian. Engenharia de software. 2ª edição. São Paulo, Brasil. Editora Campus, 1998.

[10] SALTON, Gerard; McGILL Michael. Introduction to Modern Information Retrieval. 1st ed., New York, USA, McGraw-Hill, 19