A Requirements Elicitation and Analysis Aided by Text Mining

Clay Palmeira and Rafael Chaves and Hamilton Cavalcante and Eloi Favero,

Universidade Federal do Pará, UFPa, Brasil

Summary
In Requirements Engineering the initial activities of elicitation and analysis are essential for software development. Achieving these requirements (and describing then in natural language), together with simple and rapid modeling (with enough degree of formality) are basic needs that influence the remaining stages of software development. In this paper, we provide an outline of proposal and means of validating an automatic extractor tagged text, called AutoMap, which is designed to generate concept maps on the CMapTools tool, for user requirements and described in natural language. The concept maps (CMs) are used as a modeling language requirement and have been evaluated with completeness and correctness in accordance with the requirements

Key words:
Extractor, maps, language, requirements, software.

1. Introduction

Elicitation and requirements analysis are essential for the development of software systems, and are described in the main models used for the development of software (e.g. Cascade and Incremental). Their importance is also underlined in Model of Quality of Brazilian Software Process [23]. In addition, owing to an expected result in GRE 1 sentences, the suppliers must have an understanding of these requirements.

In the elicitation and analysis, data are obtained that document the requirements for generating a code in a programming language by making necessary changes. These two activities usually occur consecutively [26]: first they discover the requirements and, following this these are documented by using some notations, either with or without the aid of a modeling tool.

One of the most common sources is documenting the requirements in a natural language, to provide an example for the user stories [4]. However, natural language can lead to ambiguity of interpretation, thus complicating the modeling of requirements, and this can cause errors and uncertainties that will be passed on to subsequent development activities.

Moreover, Concept Maps (CMs) [19], allow a representation to be made of the semi-structured knowledge of an application domain, and provides significant support in overcoming the problems arising from the use of natural language for describing software requirements. In some studies such as [17][8], CMs are used to represent the software requirements, but none of these does this automatically by using documents as a source of natural language requirements.

For these reasons outlined above, the purpose of this paper is to draw on certain resources to improve the work of [21]. This involves creating a tool called AutoMap, which serves to carry out text mining tagged with a person's lexical ID, such as: i) name, ii) verb iii) adjective, among others. This is proposed by adopting a structured approach to lexical-morphological composition of tuples in the NAME-VERB-NAME format with the objective of constructing CMs to meet extraction and modeling software requirements from sources cut in natural language texts.

This paper is structured as follows. Section 2 discusses the theoretical reference-points and is divided into three factors: i) automatic extraction of text and its complex, ii) the definitions of CMs, their applications and representations; iii) the relationship between CMs and the specification requirements. Section 3 discusses related work. Section 4 deals with the extraction methodologies and the experiments. Section 5 describes the applicability of the proposed requirements analysis. Section 6 brings the study to a conclusion and future work in this area is recommended in Section 7.

2. Theoretical References

2.1 Text Mining

A suitable approach to tagging words is well described in [21]. The basic task of labeling a sentence with words of morpho-syntactical tags is to identify them as parts of speech (nouns, verbs, etc.) and may also contain refined attributes of each category, e.g., the number and gender of a noun. The labeling often causes the problem of ambiguity in a sentence, which is usually resolved by making a reference to the context in which the word is found. The approaches are based on the following criteria: i) rules, ii) probabilistic factors, iii) hybrid cases, which combines the first and second of the criteria.

The advantage of extracting text is that it can be considered to be a way of simplifying retrieval sentence of CM construction. This observation is made by [20], who

Manuscript received June 5, 2012
Manuscript revised June 20, 2012

notes that the complexity of the tools that create the CMs, means that the users experience difficulties in using these tools. Can these circumstances can be seen in the work [7][15][9][10].
However in addition to the automatic extraction of the text, it is necessary to mine information obtained through them, and see the importance of this in the light of the problem which usually involves a complex analysis of the lexical items obtained. One of the problems of this approach is dealing with the input of the text when it is misspelled and lacks contextualization for their synonyms, as pointed out in [14].

2.2 Conceptual Maps (CM)

According to the concept defined in [19][18], the CMs can be broadly described as just a diagram showing relationships between concepts. More specifically, they can be interpreted as a set of diagrams that seek to reflect the conceptual organization of a body of knowledge or part of it. That is, their existence is derived from the conceptual structure ranging from traditional knowledge to cognitive knowledge [16].
The applicability of the CMs is aimed at teaching and/or learning processes [18] and involve a continuous analysis of what was observed in [18]. Another possible use of CMs is in the assessment of learning, not for the purpose of testing knowledge, but to classify it in some way and to learn more about the type of structure that the user has for a given set of concepts. In doing this, the CMs carry out a subjective evaluation [13], where emphasis is laid on user structure, and the CM ranks, differentiates, describes, discriminates and integrates concepts of a unit of knowledge.
The CMs are represented by employing tree methods and simple or complex graphs [18]. The tree method consists of a hierarchical representation which is usually illustrated in a binary order starting from the generic concept, which rises to a higher level by refining the lowest level. The graph representation can be formed of a simple graph that contains a single direction or a complex graph where there are bi-directional relationships. It is worth noting that a CM is a dynamic structure, and reflects the understanding of the person who uses it at a particular time [18].
The integration of knowledge is one of the main steps taken in the creation of CMs, as well as the proposed creation of an environment (as seen in [1]), to determine whether or not the end user can understand the information proposed in a CM. This can also be seen in [3], which is a study that has focused on a ‘relationship’ concept, although it also seeks to improve our understanding of the Requirements Analysis as a process of software development which can result in a better validation.

2.3 Conceptual Maps and languages of the specification requirements

There are some languages for specifying requirements that are mainly concerned with reducing the problems of natural language. In [27], the following are some examples of specification requirements for languages: Problem Statement Language (PSL) [34,33].[1], Structured Analysis And Design Technique (SADT) [12, 30].[2], Edda [35].[2], Systematic Activity Modeling Method (Samm) [17,27,32].[2], Higher Order Software (Hos) [14,15][2], and Requirements Statement Language (RSL) [23, 4][2]. In this work we opted for the use of concept maps as a language for specification.
As noted in [27], a set of desirable characteristics can be defined for the requirements specification languages should have. Thus, the purpose of the CMs can be evaluated in accordance with these characteristics and an analysis conducted of their applicability as specifications for language requirements. The characteristics and analysis of these CMs metrics are outlined below:

[a]. **User Familiarity of the specification language**: determining if the specification language can be regarded as familiar, as a result of its acceptance and popularity among developers, and has the most desirable characteristics.

Analysis: There are some works that use the CM as a requirements specification language (as seen in [17][8]), but it is not yet widely used in the market in this way.

[b]. **Textual language**: representing the requirements either using it as a natural language or as a pseudo code.

Analysis: The CMs are provided with a textual representation that defines the concepts and relationships between the concepts. e.g., i) paradigm - has - problems, ii) normal science - resolves - problems

[c]. **Graphic language**: Having a relatively small number of symbols to represent the requirements:

Analysis: CMs have basically two symbols for concepts and relationships. Some publishers of CMs offer two further graphics [6]. For a textual description of the concepts, the second refers to a CM at a lower hierarchical level.

[d]. **Hybrid Language**: this allows both graphical and textual representation, with an exact one-to-one matching relationship between the two forms of representation.

Analysis: CMs can be easily converted from a one-to-one graphic textual representation and vice versa.

[e]. **Separation of logical from physical characteristics**: the logical characteristics (whether essential or conceptual) must be represented independently of their implementations:

Analysis: Although the CMs do not have elements in their particular syntax for representing logical and physical characteristics, these may be represented because they are forms of knowledge about the system, and CM is a tool for the representation of knowledge [17].

[f]. **Multi-level abstraction**: this allows the system to be conceptualized at the highest level of abstraction, and then continually converted into lower levels of abstraction.

Analysis: CMs can have a hierarchical organization [19][17] and the most basic concepts that form a body of knowledge can be examined in detail.

[g]. **Modifiability**: the specification requirements should be structured so that requirements can be easily added or modified to meet the needs of our users or changes in environment.

Analysis: CMs are easy to change.

[h]. **Formalism**: The specification requirements must be expressed in a notation that only has a single interpretation and there must be a formal framework. If this is the case, the definition and maintenance requirements can be supported by a computer.

Analysis: With regard to this, it can be started that through the formal definition of a graph which is an ordered triple (N, A, g), the element g is a function that assigns to each arc A xy an unordered pair of nodes N, called the ends. This results in a single interpretation of two requirements.

[i]. **Rigour**: It must be ensured that the language specification requirement is supported on a mathematical basis.

Analysis: CMs can be represented by graphs and trees, thus these theoretical and mathematical diagrams can be used.

[j]. **Support of Different Development Situations**: The specification requirements must be transformed into other models used in other phases of software development.

Analysis: CMs are transformed into class models in [17][8] but they cannot be transformed into models that express temporality and sequence (e.g., State Transition Diagrams and UML Activity Diagrams).

[k]. Formalism Transparency: There should be a "one-to-one" match between the mathematical formalism and "user-friendliness" for the developer.

Analysis: CMs can be treated either in a mathematical notation of graphs as a visual representation that contains a few simple graphics.

[l]. **Independence of Design and Implementation**: a specified requirement should be independent of the design phase of implementation. It must define "what to do" rather than "how."

Analysis: same explanation is needed of the [e] "Separation of logical characteristics of the physical." feature.

[m]. **Traceability between Specification and Target Systems**: Elements in the specification should be traceable to the design and implementation.

Analysis: The translation of the CMs into class diagrams [17][8]. This shows that there can be a transformation of the part to the design stage, without having to take account of the other diagrams used in this phase (e.g., UML Activity Diagrams), but some parts from the class diagram can be generated by encoding the program.

3. Related Work

With the spread of the learning model employed by the CMs, several studies have proposed various methods for extracting text and created the CMs, from automatic and semi-automatic manual methods. However in the context of this proposal, only studies that approach the semi-automatic and automatic extraction of texts were analyzed.
Reference [15] was the first study to refer each word with a space so that a term and concept can be formed with 1, 2 or 3 terms. The user employs 1-5 weights for the text classification and to carry out some processing for standardization, such as: i) making the text upper-case ii) removing punctuation and special characters; iii) using heuristics to extract malformation, etc. It was observed that the method is relatively complex and is not always clear. A good deal of analysis is required by the user to define the quality of the CM. It displays the graphics with low quality and does not show the output of the text extracted automatically.
It [12], there is an illustration of the characteristics of an approach designed for building (semi) automatic concept maps. These were grouped into different perspectives and were related to the objectives and the data source used by the research. This highlights the fact that text lexical text extraction requires external data and databases and as a result is not as efficient. On the other hand, during the process the focus of CM is on directed and not generic texts, and thus it has to relate the previous issues. This study did not show any tool, software development or techniques for text extraction but only expressed an intention to make a CM in a semi-automatic way.
Another approach that can be adopted for automatic extraction is the use of text mining. This approach [14], involves using a tool for the creation of courses where the content is organized into CMs. The construction of the CMs is aided by the use of an algorithm based on text mining techniques to extract concepts, terms and connectors from the reference documents of a course. The extraction of texts occurs in several stages like i) parsing, ii) pattern recognition, iii) semantic analysis iv) "tokenization". It requires high throughput and heuristics,

however it does not show the output obtained by extraction.
One concern regarding the extraction of texts is the Natural Language Processing (NLP) which is related to the method of how words are treated in a self-extracting technique. An interesting approach is seen in the work of [2], which addresses the problem of the correction of words and semantics and comments on the available techniques regarding mining models associated with text database, such as WordNet [1]. This works with words through a distance vector to find similarity. However it does not use an own method, but cites other publications and the methods used by them.
Extractions can be based on automatic lexical-morphological analysis of the NLP which is observed in [5]. Its prototype tool to extract text makes use of parser PALAVRAS[2] that performs the steps of tokenization, the lexical-morphological process and conducts a syntactical analysis. In the extraction of the text, it employs tuples in the subject-verb-object format and assigns the verb the task of establishing the relationship between concepts. However, it uses ready-made tools to perform the automatic extraction of the text.

4. The Methodologies of automatic text extraction and the experiment

In reference [21], the process of making definitions begins with the selection of lexicon text in Brazilian Portuguese that contains the setting in which to retrieve tuples. In our labeling, we employed some programs developed in Java, but all the steps run locally with low computational cost and response time.
The steps required are described in [21].: i) pre-processing1 which identifies and collects/separates words dealing with pronouns, prepositions, etc.; ii) pre-processing2 which separates the words of the text input and format required by the Tree Tagger iii) executes the Tree Tagger with parameters of the Portuguese "portugues.par" which contains a lexicon of 28 million words of Corpus CETENFolha that can label the previous file in the format of one word per line iv) comparison and calculation of accuracy: comparing pairs of words-labels obtained with others that have been previously labeled and identifying the corresponding identical pairs (hits) and the different pairs (errors). The accuracy is calculated as the sum of all the hits/errors divided by the total word-pair labels submitted to labeling.
The final extraction stage of tuples is by implementing a Java class which has to be edited to enter the name of the output files. After that, it requires the user to build and run through the command prompt. In the original work [21], where the text included was tagged with 223 words or 186 words without punctuation marks, the algorithm only had one outcome in the NAME-VERB-NAME (NVN) format and none in the VERB-NAME-VERB (VNV) format, which were used to assess the degree of accuracy.
In this work that is concerned with optimizing the algorithm for automatic text extraction the resulting Java class found in the same scenario outlined, 14 tuples in the NAME-VERB- NAME format and 3 tuples in the VERB-NAME-VERB format, there is an optimization resulting in 1400% for the NVN tuples and 300% of the tuples in the VNV format. It should be pointed out that the average processing time of the original version for the optimized version of this work has on average increased by 0.8 seconds, so there was no significant change in processing despite the increase in performance.
It should be noted that the performance of the tool is obtained by an equation that represents its algorithmic complexity, where each instruction l model has an instruction time t(l). Thus, to implement the program P, for some fixed input, the instructions are processed as follows: r_1 type l_1, l2 type r_2 instructions, instructions like l_m,..r_m then the runtime of the program is given by Eq. (1)

$$\sum_{j=1}^{m} r_j - t(l_j) \tag{1}$$

5. The applicability of the requirements analysis

The experiment was based on the text from the book 'Project Database' by Carlos Alberto Heuser [11] (Page 69 Exercise 3.6: Case Study - video store), because the text clearly represents a specifications requirement. We will evaluate the CM generated by CMapTools from the output of AutoMap and analyze the results in accordance with the completeness and correctness of the requirements of the case study cited. In [22], it is stated that with the Completeness checks the user has all the requirements necessary for the specification of the system and where [24]. The means for measuring the completeness is by assessing the number of correctly identified requirements. In determining the correctness in [22], the emphasis is on the requirements that describe the functions correctly and [24]. It can be stated that correctness is the number of

---

[1] This is an electronic database consisting of lexical elements, which is arranged semantically and contains verbs, adjectives and adverbs

[2] This is part of a group of parsers VISL project (Visual Interactive Syntax Learning), the Institute of Language and Communication at the University of Southern Denmark. The parser takes as input the set of sentences of a corpus, and generates the parsing of sentences.

properly described requirements. Table 1 show the requirements observed [11] and its output is obtained by the automatic extractor, AutoMap. Since AutoMap only works with nouns and verbs, the adjectives present in requirement 2 take the meaning of the phrase. However when we look at the texts that conform to the principles of completeness, (by making a list of ten requirements that represent the problem domain), the output generated by the AutoMap proves to be very successful with a percentage of 90%, as only one of the 10 requirements, (the second) was not complied with.

There are limitations to the cardinality of the relationship between lexical elements, e.g., requirements 3 and 4, which only determines the relationship between film and tape, but cannot express the obligation (minimum cardinality equal to one) that "for each film there must beat least one tape" or express maximum cardinality, "each tape contains only a movie", in this case the maximum cardinality is equal to one. This limitation implies the unsuitability of the correctness requirements.

According to the analysis carried out by Heuser, the entities of the problem would be: i) rental ii) film; iii) tape; iv) client; v) category and vi) actor, all of which were analyzed by AutoMap extractor in the NAME lexicon. Following this, the relationships are identified as follows: i) between FILM and TAPE; ii) LOAN (RENT) between FITA and CUSTOMER iii) between FILM CATEGORY and iv) STAR between ACTOR and MOVIE. The extractor successfully identified the entities and relationships, with just one mistake, because it failed to identify the entity CATEGORY (requirement 2) and the relationship with this term, which supports the validation of the method and its accuracy.

Table I: The Requirement and the OutPut

#	*Requirement on text*	*Output by AutoMap*
1	"..Cada fita possui um número.."	Fita – possui – número
2	".. cada filme recebe um identificador próprio. Para cada fita..seu título e sua categoria"	Filme – recebe – fita
3	".. Para cada filme há pelo menos uma fita.."	Filme – há – fita
4	".. e cada fita contém somente um filme."	Fita – contém – filme
5	"… filmes estralados pelo seu ator predileto.."	Filmes – estrelado – ator
6	"..informação dos atores que estrelam em cada filme.."	Atores – estrelam – filmes
7	"..clientes cadastrados podem alugar fitas."	Clientes – alugar – fitas
8	".. locadora possui muitos clientes cadastrados.."	Locadora – possui – clientes
9	".. cada cliente recebe um número de associado.."	Cliente – recebe – número
10	".. um cliente pode ter várias fitas.."	Clientes – ter – fitas

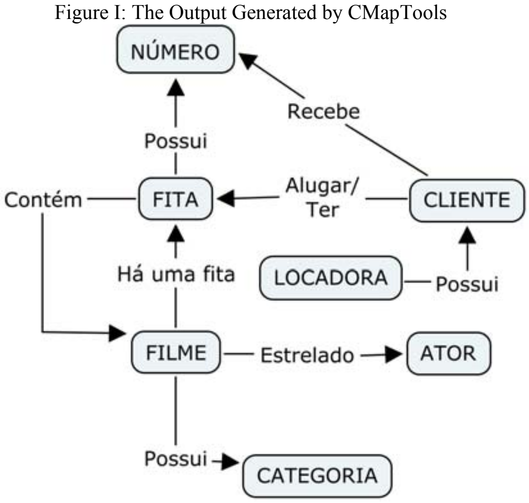
6. Conclusion

Conducting an Elicitation and Requirements Analysis is an important step in software engineering and generally requires a lot of time from the Requirements Engineer. The purpose of this study has been to prove the validity of extracting automatic concepts from texts, composing tuples in the NAME-VERB-VERB format and achieve a "hit rate" of 90%.

The CM is built by the outputs generated by the extraction tool as can seen in Figure 1, show the requirements described in natural language with an excellent level of completeness, help requirements engineers to have a better understanding of the problem domain and facilitate communication with stakeholders. The CM becomes the most useful documentation system, and as it is automatically generated, does not overload the work of requirements engineers.

However, the generated CMs do not have complete correctness; they are unable to capture the cardinality of relationships between concepts, thus requiring the cardinalities be manually assigned to undertaken by the requirements engineers. It is believed that the task of assigning cardinalities will not be significantly long and complex, which thus makes the cost-benefit ratio of CM satisfactory.

Figure I: The Output Generated by CMapTools



7. Future Works

This should make it easier to include more lexical elements in the composition of tuples; currently the work involves only one element. In a future study, there should be more lexical elements, so that the article can make sense of cardinality, and this makes it likely that the tuples will be kept closest to the formation of the concept necessary for the construction of the CM.
In addition, we are seeking to obtain the frequencies of the words of the text, as well as developing a graphical interface to make the extraction process easier and embed algorithmic complexity in the system to avoid subjecting the end user to technical difficulties. We can obtain an interactive multimedia, as seen in [25], by adding more functionality to the interface.
Other experiments will be performed to compare the CM generated by the tool with the MCs created by requirements engineers with different levels of expertise. It is also planned to evaluate the accuracy of the CM graduating in relation to their expertise.

References

[1]. Ana Marina Teixeira Araújo, Crediné Silva de Menezes, Davidson Cury (2002) "Um Ambiente Integrado para Apoiar a Avaliação da Aprendizagem Baseado em Mapas Conceituais". Departamento de Informática – Universidade Federal do Espírito Santo (UFES). XIII Simpósio Brasileiro de Informática na Educação – SBIE – UNISINOS 2002.

[2]. Anna Stravrianou, Periklis Andritsos, Nicolas Nicoloyannis (2007) "Overview and Semantic Issues of Text Mining". SIGMOD Record, September 2007 (Vol. 36, No. 3)

[3]. Bradley Hartfield (1991) "The Relation of Design to Human Concerns". Apple Computer. CH2961-1/91/0000/0494$01.00 1991 IEEE.

[4]. COHN, Mike. Desenvolvimento de software com Scrum – Aplicando Métodos Ágeis com Sucesso. Ed. Bookman. 2011. 496 p.

[5]. Cláudia Camerini Corrêa Pérez, Renata Vieira (2005) "Mapas Conceituais: geração e avaliação". Programa Interdisciplinar de Pós-Graduação em Computação Aplicada (PIPCA) – Universidade do Vale do Rio dos Sinos (UNISINOS). XXV Congresso da Sociedade Brasileira de Computação.

[6]. IHCM CmapTools – Institute for Human and Machine Cognition (IHCM), disponível em (http://cmap.ihCM.us/) acessado em 02 de abril 2012

[7]. Evangelia Gouli, Agoritsa Gogoulou, Kyparisia Papanikolaou & Maria Grigoriadou (2004) "COMPASS: AN ADAPTIVE WEB-BASED CONCEPT MAP ASSESSMENT TOOL". Concept Maps: Theory, Methodology, Technology. Proc. of the First Int. Conference on Concept Mapping. A. J. Cañas, J. D. Novak, F. M. González, Eds. Pamplona, Spain 2004

[8]. Genessa Robinson, Marcelo Scopel, Letícia Rafaela Rheinheimer, Junior Martins, Sérgio Crespo C S Pinto, Lúcia M. M. Giraffa, Crediné Menezes (2004). "Modelando Requisitos Especificados com Mapas conceituas através da UML-CM". XV Simpósio Brasileiro de Informática na Educação – SBIE – UFAM 2004.

[9]. Harry Kornilakis, Kyparisia A.Papanikolaou, Evangelia Gouli, Maria Grigoriadou (2004a) "Using Natural Language Generation to Support Interactive Concept Mapping". Department of Informatics & Telecommunications, University of Athens.

[10]. Harry Kornilakis, Maria Grigoriadou, Kyparisia A.Papanikolaou, Evangelia Gouli (2004b) "Using WordNet to Support Interactive Concept Map Construction". Department of Informatics & Telecommunications, University of Athens, Greece

[11]. HEUSER, Carlos Alberto. Projeto de Banco de Dados. 4ª Edição, número 4. Porto Alegre: Sagra Luzzatto, 2001. Cap. 3 p. 69-70 e Cap. 7 p. 172-174.

[12]. Juliana H. Kowata, Davidson Cury, Maria Claudia Silva Bôeres (2009) "Caracterização das Abordagens para Construção (Semi) Automática de Mapas Conceituais". Universidade Federal do Espírito Santo (UFES). XX Simpósio Brasileiro de Informática na Educação (2009)

[13]. Kazuya YAMASAKI*, Hiroyuki FUKUDA, Tsukasa HIRASHIMA, & Hideo FUNAOI (2010) "Kit-Build Concept Map and Its Preliminary Evaluation". Graduate school of Engineering, Hiroshima University. S. L. Wong et al. (Eds.) (2010). Proceedings of the 18th International Conference on Computers in Education. Putrajaya, Malaysia: Asia-Pacific Society for Computers in Education.

[14]. Luiz Claudio Duarte Dalmolin, Silvia Modesto Nassar, Rogério Cid Bastos (2009a) "Extrator de Conceitos e Termos Conectores para Criação de Cursos Baseados em Mapas Conceituais". Instituto de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)

[15]. Luiz Claudio Duarte Dalmolin, Silvia Modesto Nassar, Rogério Cid Bastos, Gustavo Pereira Mateus. (2009b) "A Concept Map Extractor tool for Teaching and Learning. Federal University of Santa Catarina (UFSC). 2009 Ninth IEEE International Conference on Advanced Learning Technologies

[16]. Luiz Claudio Duarte Dalmolin, Silvia Modesto Nassar, Rogério Cid Bastos (2008) "Uma ferramenta de mineração de documentos para a construção de mapas conceituais". Instituto de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC). V Congresso Brasileiro de Ensino Superior a Distância, Gramado. (2008).

[17]. Luiz Henrique Martins Lins Rolim (2006) "Utilização de Mapas Conceituais em Engenharia de Software: Projetando uma ferramenta case". Anais do 12O Encontro de Iniciação Científica e Pós-Graduação do ITA – XII ENCITA / 2006 Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, Brasil, Outubro, 16 a 19, 2006

[18]. Marco Antonio Moreira (2006) "MAPAS CONCEITUAIS (Concept maps)". Instituto de Física,

UFRGS. Trabalho utilizado em um "workshop" sobre mapas conceituais oferecido no Segundo Congresso Internacional sobre Investigação em Didática das Ciências & das Matemáticas, Valência, Espanha, 23 a 25 de setembro de 1987. Adaptado de uma conferência proferida na Terceira Reunião Nacional de Educação em Física, Córdoba, Argentina, 5 a 8 de outubro de 1983. Publicado em CONTACTOS, México, 3(2):38-57, 1988, em Monografias do Grupo de Ensino, Série Enfoques Didáticos, N° 2, 1991 e na Série Textos de Apoio ao Professor de Física, N° 3, 1992. Revisado, atualizado e ampliado em 2006.

[19]. Marco Antonio Moreira (1997) "MAPAS CONCEITUAIS E APRENDIZAGEM SIGNIFICATIVA" Instituto de Física – UFRGS. Adaptado e atualizado, em 1997, de um trabalho com o mesmo título publicado em O ENSINO, Revista Galáico Portuguesa de Sócio-Pedagogia e Sócio-Linguística, Pontevedra/Galícia/Espanha e Braga/Portugal, N° 23 a 28: 87-95, 1988.

[20]. Mary Beth Rosson, Hansa Sinha, Mithu Bhattacharya and Dejin Zhao (2007) "Design Planning in End-User Web Development". The Pennsylvania State University. 2007 IEEE Symposium on Visual Languages and Human-Centric Computing

[21]. Miriam L. Domingues, Eloi L. Favero, Ivo P. Medeiros (2007) "Etiquetagem de Palavras para o Português do Brasil". TIL – V Workshop em tecnologia da informação e da linguagem humana. Rio de Janeiro (2007).

[22]. Miriam Sayão, Arndt Von Staa, Julio Cesar Sampaio do Prado Leite (2003). "Qualidade em requisitos" Monografias em Ciência da Computação nº 47/03. ISSN 0103-9741. Departamento de Informática. Pontifícia Universidade Católica do Rio de Janeiro (PUC-RJ). Outubro (2003).

[23]. MPS.BR: SOFTEX: MPS.BR – Guia Geral, Versão 2011, MPS.BR:2011. [Online]. http://www.softex.br/mpsbr acessado em 02/04/2012.

[24]. Rodrigo Cezario da Silva (2011) " Uma abordagem para o reuso de requisitos baseada em padrões e rastreabilidade" Dissertação de mestrado em computação aplicada. Univali (2011).

[25]. Romero Tavares (2007) "Aprendizagem significativa em um ambiente multimídia". Departamento de Física – UFPB. Indivisa, Boletín de Estudios e Investigación, 2007, Monografía VIII, pp. 551-561. ISSN: 1579-3141

[26]. SOMMERVILLE, Ian. Engenharia de Software – 9a edição. Ed. Pearson Education –BR, 2011. 568 p

[27]. T.H. Tse, L. Pong (1991) "An examination of requirements specification languages" The Computer Journal (1991) 34 (2): 143-152. doi: 10.1093/comjnl/34.2.143

Clay Palmeira received graduated in Data Processing Technology in the University Center of Pará (2000). He specializes in Analysis Systems from the Federal University of Pará (2002) and Master Student in Computer Science from UFPa (2011). Develop software using the principles of Artificial Intelligence (AI) in the area of Genetic Algorithms (GA's). Has experience in Computer Science and Computer Networks with emphasis on Analysis of Algorithms and Complexity of Computing, acting on the following topics: software development, artificial intelligence, neural networks, internet, development, e-commerce, computer networks and their relationships. Currently is part of group GERCOM UFPa which conducts research in the area of Delay Tolerant Networking (DTN) associated with AI techniques.

Rafael Chaves holds a Masters in Computer Science from University Federal of Santa Catarina (2003). He is currently Assistant Professor III of the Federal University of Para, and PhD from the Graduate Program in Electrical Engineering, described his thesis proposal in September 2011. He has experience in Computer Science, working mainly on the following topics: teaching software engineering, process simulation software, templates and process quality MPS.BR

Hamilton Cavalcante is student of Bachelor of Computer Science, Federal University of Pará Seeking to gain experience and to enable the new professional reality. Available for work experience, scientific research, travel, events and courses. Are you interested in the Database.

Eloi Favero is graduated in Bachelor in Computer Science from University Federal do Rio Grande do Sul (1987), Masters in Computer Science from University Federal do Rio Grande do Sul (1990) and Ph.D. in Computer Science from Federal University of Pernambuco (2000). He has experience in computer science, with emphasis on Basic Software, acting on the following topics: automatic evaluation, learning environment, distance learning environment, assessment of learning and intelligent agents.