Restructuring Unstructured Documents

On the use of smart and semi-automatic interfaces to structure unstructured data

Jacques Péré-Laperne[1, 2]
[1]1A3I
F-64210 Bidart, France
email : j.perelaperne@gmail.com

Nadine Couture[2]
[2]ESTIA, Univ Bordeaux, LaBRI, UMR 5800
F-64210 Bidart, France
email: nadine.couture@u-bordeaux.fr

***Abstract*— Every day, the volume of the world's digital data increases considerably. Over 75% of these data are non-structured. This paper is about restructuring graphic information contained in Portable Document Format (PDF) files and/or vector files. These documents are generally held by ''Smart Factory'' services: design offices, methods departments, new work departments and company maintenance services. To restructure these data, we propose using Knowledge Discovery in Databases (KDD) methods. Although, theoretically, the user is present during the KDD, in practice, this is not the case. This was observed by Fayard in 2003 at the KDD conference. Generally, the user is only present during the validation phase. We show why, in data restructuring, the user must be at the heart of the process and present at all stages. We can talk about (A)KDD for the Anthropocentric Knowledge Discovery in Databases .The first stage of this restructuring consists of extracting graphic and text objects contained in Portable Document Format (PDF) files to put them in a pivot data format. The second stage consists of coding this information in the form of an alphabet. The third stage consists of recreating the graphic and text components which are repeated in these files (which we shall refer to as graphemes). And the fourth stage consists either (1) of automatically identifying these graphemes based on knowledge or (2) presenting them so the user identifies and introduces them into the knowledge base. It is this entire restructuring process, which we will describe in this paper. As we highlighted, in this incremental process it is people who play the main role, assisted by computers and not the opposite.**

Keywords- Data Minning; Computer Human Interface (CHI); Portable Document Format (PDF); Knowledge Discovery in Databases (KDD); Graphic reconstruction; Pattern recognition .

I. INTRODUCTION

A. Context

Currently, almost all information which is generated and circulates worldwide is digital. In 2001, Nigay [1] stated in "Modality of interaction & multi-modality", a University of Berkeley project estimated, that the quantity of data generated annually in the world was an exaoctet (1 million terabytes = 10^{18} bytes). 99.997% of these data are available in digital form''.

The International Data Corporation (IDC) predicted that these data would between now and 2020 increase to 40 zetta-bytes (ZB), = 10^{21} octets), i.e., 50 times more than in 2010 and 40 000 times more than in 2001[24].

Digital data are increasing considerably. These digital data are mostly non-structured or more exactly destructured. Destructured because, today, they are produced by software, but stored in very "poor" data format files. Computer World states that non-structured information could represent over 70% - 80% of all world data, and, 95 % of the "Big Data" [25]. These data contain a priceless wealth for the companies which own them, provided they are restructured.

B. Real industrial need

The data present in destructured files are visible, transmissible and archivable, but they cannot be used, modified, and more generally they are inoperable. Any mechanism which enables these data to be restructured and used represents an extraordinary benefit for their owners. This explains the explosion of research in the field of KDD.

Restructuring does not need to be total. Often partial reconstruction is sufficient and it is the user who will decide on the level of reconstruction.

We can take, for example, the case of an aircraft manufacturer, which has aircraft wiring diagrams. If it needs to know the whys and wherefores of equipotentials for part of the aircraft, the user will not need to restructure all the information. Knowing only several pages of connectors and equipotentials, as well as several components, is enough.

In 2003, the LORIA team from Nancy made the following observation [2]: "Grassroot users, as well as large companies, have a huge amount of information at their disposal, but this information is available in very "poor" formats: paper documents, or low-level, poorly structured digital formats such as Postscript, PDF or DXF. The challenge is, therefore, to convert this poorly structured information into enriched information which can be used within an information system".

C. Case study and application

All these digital data, which we are interested in and want to restructure, include graphic and vectorial digital data. These data come from computer-assisted drawing, computer-aided design and desktop publishing or "technical" software. The main feature of these data is the mixture of text and drawings, with the particularity that there are generally more drawings than text.

 ISBN: 978-1-61208-565-4

These graphic and vectorial data are present in technical files, plans, notices, machining sequences, etc. These data are held by design offices, methods or production departments of "Smart factory" industrial concerns.

For example, in the precise field of diagrams Process & Instrumentation Diagrams (PID), the total amount of graphic data held by Small & Medium-sized Enterprises (SME) is very small compared to the total world volume of destructured data which we mentioned above. We can talk about "Small Data" for SMEs and perhaps "Medium Data" for large companies. This is a far cry from "Big Data", which is often behind KDD.

D. Our vision

"Deep learning" algorithms or convolution neural networks [3] cannot be used in our case for 2 main reasons: the small amount of information available in the smart factory and the difficulty of creating learning data.

Our approach differs from current work. It is more graphic and places the user at the heart of the extraction process. We do, of course, use all the text present in files, but we also use all the graphic objects which materialise title blocks, connecting elements, components, functions, etc.

As we have just said, the user is present in all the restructuring phases. It is much less expensive and more efficient to get the user to intervene throughout the process (and especially at the beginning) rather than at the end to validate or correct errors. Getting the user to take decisions (which only takes a few seconds according to the Man Machine Interface implemented) considerably reduces and even cancels the error rate for all the tasks.

In the restructuring process, there are around twenty tasks to be carried out. If each task is fully automated and produces an error rate of between 2 and 3 % (which is very low, 97% to 98% exact recognition rate), we can observe that at the end of the validation chain the user will have to take decisions about information with between 40% and 60 % errors. Errors in the first tasks lead to errors in subsequent tasks.

It is the error rate which can lead the user to reject the system. In the validation phase, the user must modify, reclass or cancel over 40 % of results. If the destructured datum contains 10 000 pieces of basic information, over 4000 pieces of information will have to be modified. These correction tasks will take a considerable amount of time, which is unacceptable in the world of industry.

Our aim is to restructure "Smart factory" user-anthropocentirc technical data. This allows us to offer the user a more ergonomic interface, which we can qualify as "Smart Interface" in the "Smart factory".

In section 2, we make the state of the art, then, in section 3 we explain why the user must be at the center of the process. In section 4 we describe the principle of restructuring, and in section 5 we decompose this principle into phases, stages, tasks and sub-tasks. The issue of visualization of a great deal of information is discussed in section 6. We conclude and we precise the future work in section 7.

II. STATE OF THE ART

Since the 2000s, many teams have dealt with restructuring these destructured data.

(i) Restructuring the pages of newspapers or magazines contained in PDF files [4] [5]: Major work was carried out on extending "PDFBox" functionalities to restructure text information, and separating it from graphic information using "best-first clustering". Information is dealt with as a whole.

(ii) Analysing PDF files and extracting mainly text data in XML format by Dejean and Meunier in 2006 [6]: The aim is to recreate words, associate letters, then lines, paragraphs and articles, and then separate text from images or vector areas. Images and vector areas are also, in this case, considered as a whole.

(iii) Extracting tables from PDFs [7]: They developed several heuristics to recognise, extract and store information in tables in XML files. These heuristics are based on the absolute position of elements.

(iiii) Extracting tables and forms from PDF [8] : After having defined the notions of tables and forms, they present a set of solutions for extracting data mainly based on the proximity of elements in a 2D space (X, Y) and finish by quoting university and commercial tools.

(iiiii) Extracting knowledge from scientific papers in PDF [9]: Developed using Python, the interactive tool, enables paper headers, quotations, figures, tables and algorithms to be extracted. Images are extracted without interpretation.

(iiiiii) Extracting and identifying graphs and images from PDF files [10]: They define 16 graphemes for identifying 36 types of graphs (curves, bar graphs, etc.), from a base of 14,000 BioMed Central (BMC) papers. They use "machine learning" algorithms to identify 16 types of graph (Weka 3 and LogitBoost).

Finally, we observed that not much research has been carried out into extracting knowledge from documents with a high graphic component. The Bordeaux INRIA Mnémosyne team dealt with this issue in the paper[11] ''Implicit knowledge extraction and structuring in electrical diagrams''. Ikram Chraibi Kaadoud extracted all the text from PDF files, then used methods inspired by KDD to obtain the information required to structure knowledge. This knowledge was modelled with dendrograms.

III. THE ESSENTIAL ROLE OF THE USER IN THE RESTRUCTURING PROCESS

In the abstract, we described the main stages of this restructuring. These stages are the same as (KDD) described by Fayard in ''From Data Mining to Knowledge Discovery in Databases. Advances in Knowledge Discovery and Data Mining''[12].

In 1996 Fayyad insisted on the fact that knowledge should be extracted with the user's help. At the 2003 KDD conference, he observed that a great deal of KDD work is carried out without the user. The user only intervenes in the final phase [13].

The same observation was made by Chevrin et al. [14]. They insisted on the user's role and this was one of the first

times the term ‘‘Anthropocentric’’ could be seen to take on such importance in KDD.

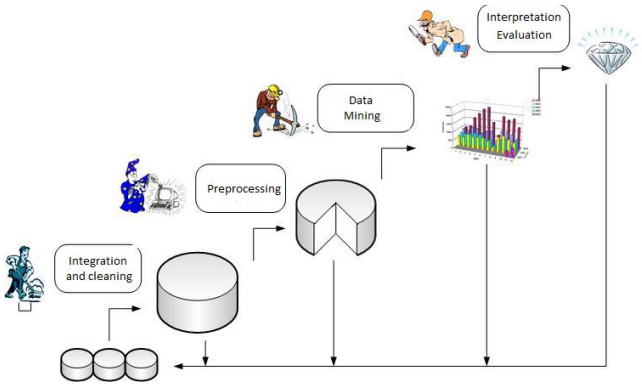

Figure 1. The KDD view by Chevrin [14]

We also made this observation: in most of the work we have quoted with regard to restructuring destructured documents, the user is present (sometimes very little). However, this work is not "Anthropocentric", based on the user.

This observation can still be seen today in a large number of works on knowledge extraction. Most of the time, the aim of research is to automate this extraction as far as possible. This is certainly possible when there are a lot of data to be processed, but in our case it is impossible. The user wishing to restructure these data does not have enough data available to fully and reliably automate the restructuring process.

Generally, in the ‘‘Smart Factory’’ it is the user who pilots restructuring with the help of the computer due to cost and efficiency, and to avoid rejection of the system.

IV. THE RESTRUCTURING PRINCIPLE

The general restructuring principle is quite simple. We search for “Graphemes” (as described by Jacques Bertin [15]) which are repeated in the same types of documents. According to how often they appear, the user analyses whether these graphemes correspond or not to notions which the user is used to handling. Once analysed by the user, these elements are integrated and parameterised in “the Knowledge Base” (true and false). In reality, “The” knowledge base is made up of a whole range of knowledge bases, each of which corresponds to a type of document or document sub-type.

After the analysis by the user of a grapheme or set of graphemes, the search for graphemes is restarted, taking into account the knowledge acquired in the previous phase. The user is presented with new proposals. The 3 “I’s” are the **I**terative, **I**nteractive and **I**ntegration aspect of KDD defined in [12].

In this case, this principle for reconstruction is Incremental (KDD’s 4th ‘‘I’’). Once basic graphemes have been recognised, “Supers Graphemes” are searched for. These are the associations of graphemes and basic graphic objects. They are, in turn, presented to the user and integrated in the knowledge base. This process is recursive until any knowledge to be searched for in the destructured documents is obtained.

The restructuring of documents is highly prioritised. It is described by an ontology which is created, or adapted, by the user based on general ontologies or parts of ontologies which act as a model.

The ontology is created throughout the KDD process.

If, for example, we take the case of a factory and its technical documents, we will find: methods documents, production documents, maintenance documents, etc. -- For maintenance documents: building plans, plans of industrial processes, fluid circulation plans, etc. -- For fluid circulation plans: hydraulic plans , pneumatic plans, electric plans, etc. -- For electric plans: building electric plans, electromechanical plans and plans of machine automatons, etc. -- For automaton plans: those created with ‘AutoCad’ software, those created with’Elec’View’ software, those created with ‘See Expert’ software.

By making the knowledge base as close as possible to the document subtype, we obtain better restructuring rates. The information presented to the user is more reliable. If the basic symbols to be restructured are the same for AutoCad, Elec’view or See Expert (all these software are CAD software), this is not the same for title blocks, automaton input and output cards, variators, specific cards, etc. These graphic data differ from one software program to another.

Likewise, objects to be restructured in architectural plans are very different from those of automation diagrams or pneumatic diagrams. Ontologies differ greatly according to the field.

V. THE STAGES OF RESTRUCTURING DESTRUCTURED DOCUMENTS

Each of the phases of (A)KDD is divided into tasks and sub-tasks so that the user can intervene and make choices. The word “User” is a general term which concerns all types of users intervening in restructuring, from experts to simple users.

“One of the key words in Interface Human Machine (IHM) is user centered. User-centred design by Norman & Draper [16] is a paradigm defining 12 key principles which place the user at the heart of development processes”. This is the paradigm stated by Dupuy-Chessa [17] which we use for user and system tasks. For reasons of clarity, we will only describe the main tasks of each of the phases in this paper.

For each of the stages and tasks we state how this task can be implemented in a real industrial data structuring process. We highlight the tasks for which, as far as we are aware, there are no efficient implementable solutions. It is for these tasks that we will show that a main system interaction loop appears to us to be the only efficient one.

A. Phase 1 of (A)KDD: cleaning and integration

To integrate data, the user must define the arborescence of data. This is this phase’s first task. This arborescence will be created in an ontology. To create the ontology we use Stanford University’s *Protégé* software. *Protégé* is the most

popular software for creating ontologies and knowledge [18], based on figures from Jorge Cardoso's 2007 study. *Protégé* is very user-friendly and simple, and works well for the needs of restructuring.

The second task is a system task. It consists of inputting different types of highly destructured file formats and extracting the following graphic information from them:

- Segments, of polylines
- Text from 1 to n characters
- Pens, brushes and fonts.

When more complex graphic objects are present, like circles, circular arcs, ellipses, squares, rectangles, triangles, etc., these objects are transformed into polylines.

This graphic information is put into a pivot format. We chose the Autocad format (DXF/DWG). This format is readable by the main computer-aided design, drawing and desktop publishing software. Input file formats include: Portable Document Format (PDF), PostScript (PS), Enhanced MetaFile (EMF), Metafile (WMF), Computer Graphics Metafile (CGM), Scalable Vector Graphics (SVG, SVGZ), AutoCad formats (DXF, DWG), Hewlett-Packard formats (HPGL, HPGL2, PCL), Initial Graphics Exchange Specification(IGES), Standard for the Exchange of Product model data(STEP), Stereolithography file format (SLT), etc.

This system task is more or less complex according to the type of input files. It is essential for data restructuring. It is a pure engineering task.

The third task is for the user to select the files or parts of files to be processed. A tree based on the ontology model and a highly interactive file viewer are used to help the user choose.

B. Phase 2 of (A)KDD: Pre-processing

When we analyse PDF files and vectorial files from the same software, we can see that graphic objects are always drawn in the same chronological order. This is true for basic graphic objects, squares, rectangles, etc., and also more complex graphic objects, such as title blocks, automaton cards, etc. Generally, this chronological information is adjacent. For some software, this is not the case. We first of all find segment type graphic objects, then a little further in the file letter type graphic objects.

Most solutions, to restructure information, use the X and Y positions of objects, and their closeness. This is a solution which we use, but we give priority to the chronological nature of data. To achieve this, we code all the basic graphic objects (like for an alphabet). We can therefore handle "character" (code) chains instead of 2D graphic objects. Search times are reduced considerably.

We use 2 types of coding. Firstly, coding which takes into account the length, position and orientation of objects in a point of reference X,Y which originates at the bottom left of the page. Secondly, coding which codes an object according to the previous object; this way our coding is not sensitive to translation, rotation or a change of scale. A very simple example of coding could, for long segments, be 'V' for Verticals, 'H' for Horizontals, 'O' for Obliques, and 'h', 'v', 'o' for short segments. This is the principle we chose with greater distinction for lengths and angles.

The first system task consists of determining the density with which segments are distributed according to their lengths and orientations.

The second task presents the results to the user via a very visual and ergonomic interface to make a choice. The coding thus implemented will be more or less extensive.

C. Phase 3 of the (A)KDD: Data mining

Data mining for restructuring can be broken down into several stages, which the Incremental part of (A)KDD:

Stage 1: Restructuring dotted and dash lines: This involves defining the types of lines and identifying them based on their composition (length of features and spaces).

Stage 2: Restructuring letters: (some software directly draws the letters of words, especially for tracer files). This involves defining the font, size, style, orientation and letters which associate one or several scripts, eg. "e", "é", "ë", a type of font (Arial, Courrier, ISO, etc).

Stage 3: Restructuring words: This involves grouping together identified letters to form words, then text (sets of words). For assembly notices we have to identify phases and paragraphs.

Stage 4: Restructuring basic symbols: This involves defining these symbols and associating basic graphic elements and text.

Stage 5: Restructuring complex symbols: This involves defining these symbols and associating graphic elements from identified text and symbols. This stage is recursive.

Stage 6: Restructuring functions: This involves associating simple and complex symbols and basic graphic elements which make up a function. This stage is also recursive, a function is made up of sub-functions, itself made up of sub-functions, etc.

Stage 7: Restructuring the file: This involves defining the major fields and sub-fields in which we group together functions. It is also recursive.

Each of the stages can be broken down into 2 main tasks.

The first task defines the present notions. It is the user who introduces them into the knowledge base's ontology or duplicates them from close knowledge base ontologies.

The second task is a system task. It consists of identifying graphic patterns which are repeated the most in files. This identification is based on "Stringology" methods which we apply to the 2 codifications of the graphic elements described above and which we validate with positions X and Y. Throughout restructuring these codifications evolve and the recognised graphemes are automatically replaced by a new code associated with this new grapheme.

The main techniques of stringology that we use are as follows: Creating "chain tables"; Searching for one or several chains among a set of chains; Regular expressions.

D. Phase 4 of (A)KDD : Evaluation & presentation

For each stage of the previous phase 3 it is the user who evaluates and validates the information presented. This phase 4 can be broken down into 3 main tasks. Firstly, a system/user task which presents all the information from a system task for the stage in question, enabling the user to interact. Secondly, a user task which selects and introduces information into the

knowledge base. Thirdly, a system task which recalculates all the information taking into account the identified "object", and loops back on the first task. It is the first task which is the core issue that will develop in the next section.

VI. How to visualise a large amount of information

Restructuring is a long and complex operation. It must be 100% exact, which is why it is user "Anthropocentric". However, the user tasks described above must be ergonomic, implicit, simple and efficient, especially those which visualise a lot of data, which are the core issue.

The problem is simple:

How to visualise as much information as possible on a station which will be used by the "Smart factory's" technical work services over the next 10 years?

How to navigate through this large amount of information? We have between 100 and 200 pattern to recognize in a document, and for each pattern we have between 1 and 10 proposals.

How to identify the information that we want to restructure because everything is not to restructure?

How to choose which of the information displayed and grouped by pattern similarity corresponds exactly to the element to be restructured?

How to introduce it into the knowledge base?

If the problem is simple, the solution is more complex. It can be seen that each of these questions involves system and user sub-tasks that interact with one another. The task T1 is thus divided into several sub-tasks in which the user is at the center of the action.

The issue is simple: How to visualise as much information as possible on a station which will be used by the "Smart factory's" technical work services over the next 10 years? The solution is more complex.

This is why we have focused our research on the "Overview first, zoom and filter, then details-on-demand" concept described in Shneiderman's "information seeking mantra" [20], respecting TTT taxonomy (Shneiderman 96) and its seven tasks: "overview, zoom, filter, details-ondemand, relate, history and extract" which Kleim in 2004 [1] covers in the definition of the "Visual Exploration Paradigm".

"Focus + Context" techniques, such as "FishEyesView" are very promising in certain fields, like that of rules of association [14]. However, they are not suited to our issue because the possibility for enlargement is small (4 to 5 times the original size). An in-depth study by Mikkel Rønne Jakobsen and Kasper Hornbæk shows that, on small screens (640x267) or medium screens (1920x800), "Overview + Detail" methods perform better than "Zooming + Panning", which itself performs better than "Focus + Context", even for very large screens (5760x2400).

We made the same observation, which led us to develop a combintaion of both methods: "Overview+Detail" and "Zooming+Panning". In this combination, the methods and uses can be divided into three groups, those fixed by the system, those chosen by the system and those chosen by the user.

VII. Conclusion and future work

This work is part of integrating visualisation techniques with disciplines, such as knowledge extraction, learning, handling and exploring high volumes of data.

The final aim is to contribute the power of IHM in each computer in "Smart factory" design offices to enable data to be explored better, more quickly and more intuitively. The final objective is to improve the economic performance of these companies.

What we wanted to show is that in a complex process like the restructuring of the graphic documents it is the user who must be at the heart of the system. Without the user or with a late intervention in the process, the system is not viable. Like Fayyad, we decompose the process into 4 main phases. Each phase breaks down into stages. And each step, as advocated by Dupuy-Chessa, is broken down into tasks and sub-tasks either system or user. This is how we can put the user at the heart of stage step.

Our observation is that the user must be present from the beginning to reduce errors. This forces us to design a user-friendly HMI this throughout the restructuring process.

The "anthropocentric" character of this approach is opposed to the automatic work that is to be found in many of the work around KDD. This leads us to develop this method (A) KDD with the sole objective of obtaining better results and, above all, a better efficiency, two notions necessary in the industrial world.

Our next work will aim to demonstrate that the systematic application of (A)KDD gives better results than automatic methods for complex problems.

References

[1] L. Nigay, "Modality of interaction & multi modality," In Thése habilitation diriger des recherches. préparé au Laboratoire de Communication Langagire et Interaction Personne-Systme (IMAG), Université Joseph Fourier, pp. 1-74, 2001.

[2] K. Tombre and B. Lamiroy, "Graphics Recognition – from Re-engineering to Retrieval," 7th International Conference on Document Analysis and Recognition, Aug 2003, Edinburgh, Scotland. UK, IEEE Computer Society Press, pp. 148-155, 2003.

[3] Y. Bengio and Y. LeCun, "Scaling learning algorithms towards AI," in Large Scale Kernel Machines, (L. Bottou, O. Chapelle, D. DeCoste, and J. Weston, eds.), MIT Press, 2007.

[4] T. Hassan and R. Baumgartner, "Table recognition and understanding from PDF files," in Proc. 9th Int. Conf. on Document Analysis and Recognition, Parana, pp. 1143-1147, 2007.

[5] T. Hassan, "User-guided wrapping of pdf documents using graph matching techniques," In 10th International Conference on Document Analysis and Recognition. IEEE Computer Society, Los Alamitos, CA, USA, pp. 631–635, 2009.

[6] H. Déjean and J.-L. Meunier, "A system for converting PDF documents into structured XML format," in DAS 2006: Proceedings of the International Workshop on Document Analysis Systems, pp. 129-140, 2006.

 ISBN: 978-1-61208-565-4

[7] B. Yildiz, K. Kaiser, and S. Miksch, “Automatic Table Recognition and Extraction from Heterogeneous Documents,” in Journal of Computer and Communications 03(12), pp. 100-110, January 2015.

[8] B. Coüasnon and A. Lemaitre, “Recognition of Tables and Forms,” in Doermann, D., Tombre, K. (eds.) Handbook of Document Image Processing and Recognition, pp. 647–677. Springer London, 2014.

[9] J.Wu and al., “Pdfmef: A multi-entity knowledge extraction framework for scholarly documents and semantic search,” in K-Cap, 2015.

[10] M. Shao and R.P. Futrelle, “Recognition and Classification of Figures in PDF Documents,” in W. Liu and J. Llados (Eds.): GREC 2005, LNCS 3926, pp. 231–242. Springer-Verlag Berlin Heidelberg, 2006.

[11] I. Chraibi Kaadoud, N. Rougier and F. Alexandre, “Implicit knowledge extraction and structuration from electrical diagrams,” at IEA/AIE, 2017.

[12] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, “From Data Mining to Knowledge Discovery in Databases Advances in Knowledge Discovery and Data Mining,” in AAI Press / The MIT Press, pp. 1-43, 1996.

[13] U. M. Fayyad, G. Piatetsky-Shapiro and R. Uthurusamy, ”Summary from the kdd-03 panel: data mining: the next 10 years,” in SIGKDD Explor. Newsl., ISSN 1931- 0145, vol.5, no.2, pp.191-196, 2003.

[14] V. Chevrin, O. Couturier, E. Mephu Nguifo and J. Rouillard, “User-driven association rules mining to decisions supports systems,” in revue IHM (RIHM) ISSN 1289-296, Vol.8, pp 201- 220, 2007.

[15] J. Bertin, “Sémiologie graphique, Les diagrammes – Les réseaux – Les cartes,” (Graphical Semiology, Diagrams - Networks – Map), in Paris, Editions de l’EHESS, 4e éditions, (1ère édition: Paris, Editions Gauthier-Villar, 1967), 2005

[16] R. D. Pea, “User Centered System Design: New Perspectives on Human-Computer Interaction,” in Journal educational computing research, pp. 129-134, 1987.

[17] S. Dupuy-Chessa, “Modeling in Human- Machine Interaction and Information Systems: At the Crossroads,” Interface homme-machine [cs.HC]. Université de Grenoble, 2011.

[18] N. Malviya, N. Mishra and S. Sahu , “Developing University Ontology using protégé OWL Tool,” in International Journal of Scientific & Engineering Research Vol.2, Issue 9, September 2011.

[19] J. Cardoso, “the Semantic Web Vision: Where are We?” in IEEE Intelligent Systems, pp.22-26, September/October 2007.

[20] B. Shneiderman, “The eyes have it: A task by data-type taxonomy for information visualizations,” in Proceedings of Visual Languages'96, IEEE Computer Science Press Publ., pp. 336-343, 1996.

[21] D. A. Keim, “Information Visualization and Visual Data Mining B., The eyes have it: A task by data-type taxonomy for information visualizations,” in IEEE Transaction on Vizualizationn and computer graphics, vol. 7, no. 1, January-March 2002.

[22] G. Bothorel, M.Serrurier and C. Hurter, “Using visual data minning tools to explore a set of association rules,” in RIHM Art. N° 12, 2011.

[23] M. Rønne Jakobsen and K. Hornbæk, “Sizing Up Visualizations: Effects of Display Size in Focus+Context, Overview+Detail, and Zooming Interfaces,” in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, New York, NY, USA, pp. 1451–1460, 2011.

[24] D. Cuny, “The Great Information: The New Revolution”, from a study IDC-EMC, "Extracting the Value of Chaos" by EMC Gartner, Tribune No. 42, pp. 4, March 29 to April 4 2013

[25] A. GANDOMI and M. HAIDER, “Beyond the hype: Big data concepts, methods, and analytics”. International Journal of Information Management, 2015, vol. 35, no 2, pp. 137-144.

 ISBN: 978-1-61208-565-4