Graphical Visualization in the Knowledge Management System Atanor

Bruno Pinaud, Pascale Kuntz, Fabrice Guillet, Vincent Philippé

Graphical Visualization in the Knowledge Management System Atanor

Bruno Pinaud
(Laboratoire d'informatique de Nantes Atlantique, France
Knowesia SAS, France
bruno.pinaud@univ-nantes.fr)

Pascale Kuntz and Fabrice Guillet
(Laboratoire d'informatique de Nantes Atlantique, France)

Vincent Philippé
(Knowesia SAS, France)

Abstract The interaction between a knowledge management systems and the users requires well-adapted visualization tools with graphical formalization of knowledge. The formalization is often theoretically based on graph-models. Yet, the best associated visual representations use trees but may be more limited than those with graphs. This paper gives an introduction to Atanor, a knowledge management system, whose graphical model for visualizing knowledge is tree-based. However this approach entails vertex redundancies. Consequently, we develop a new approach based on a layered digraph to solve this problem. Finally, we draw a comparison on an industrial example showing the advantages of the new model.
Key Words: Knowledge visualization, knowledge management systems, graph layout
Category: H.3.m

1 Introduction

Interactive visualization supports are taking an increasing importance in the development of the operational Knowledge Management Systems (KMS) [Burkhard, 2004]. Upstream the process of KM collecting the expertise is made much easier thanks to well-adapted visual interfaces. Downstream the process these interfaces allow a presentation of the knowledge according to different points of view, and facilitate their handling by different categories of users. To satisfy these requirements, the interactive visual representations should respect both the semantic of the information, and hide the algorithmic and technical aspects as much as possible [Eppler and Burkhard, 2005].

In this context, graphs represent interesting models. They can be used both as abstract models for inherent relations among data, and as an efficient visual representation which makes easier access to complex structures without getting bogged down in theoretical concepts [Herman et al., 2000].

In KM, most of the graph-based models are inspired from the generic model of the semantic networks [Lehmann, 1992]; the vertices represent concepts and the arcs describe semantic relations between them. Representations based on a tree structure are probably the most popular, certainly due to their apparent simplicity. We can quote firstly, the Mind Map [Buzan and Buzan, 1996] where a user can draw some ideas around a central node; then, the fault trees [Limnios, 2005] to represent all the possible problems of a system and finally the decision trees [Quilan, 1990] mainly used for sorting and classifying data. Representations based explicitly on a graph model can be put into three non-independent families: the conceptual graphs [Sowa, 1992] for visualizing relations between data and representing ontology; the ontology engineering for representing and manipulating ontology like in Os-Skill [Roche et al., 2005]; the Bayesian networks, which are state graphs with a probability of transition on each arc, used in software like ITM (`http://www.mondeca.com`). In addition, we can find some graphical representations developed specifically for the classical methods of KM [Corby and Dieng, 1998].

In this paper, we propose a new visualization framework based on a graph model for a KMS (Atanor), which is knowledge-deployment-oriented in an operational context for complex systems [Guillet et al., 2002]. For instance, the applicative framework used in this paper, is the maintenance aid for mail sorting machines. In the first version of the system, the graphical model was based on a generalization of the fault trees and the decision trees. Experiments in real-life situations have proved that this representation was not intuitive and difficult to use in the expertise description phase. As the Atanor basic ruleset is by nature structured as a graph, the tree representation entails numerous redundancies which can hide important characteristics of the systems studied. We here propose an extension to a layered graph-based model which is more adapted both to expert reasoning and information retrieval.

The rest of the paper is organized as follows: [Section 2] briefly describes the knowledge server Atanor. [Section 3] describes the tree-based model and the new graph-based model. [Section 4] draws a comparison of both models when applied to the maintenance aid for mail sorting machines.

2 The Atanor Knowledge Management System

Atanor was initially directed towards the capitalization, transmission and preservation of the useful contextual knowledge in a multimedia form easily activated [Figure 1]. These phases are articulated in the Atanor's architecture around three models: *(1)* the organic-parts model provides a physical description of the system at different granularity levels. Besides the classical component hierarchies and their descriptions in different formats (text, image), it integrates intelligent product manuals and dynamic scenarios in virtual reality. *(2)* The expert model guides the tacit-to-explicit knowledge conversion process and allows reasoning with a rule system [Figure 2]. *(3)* The competency model describes the association between the required skills for the different operating tasks and the human resources [Vergnaud et al., 2004].

These three models are strongly connected and their activation is task-centered

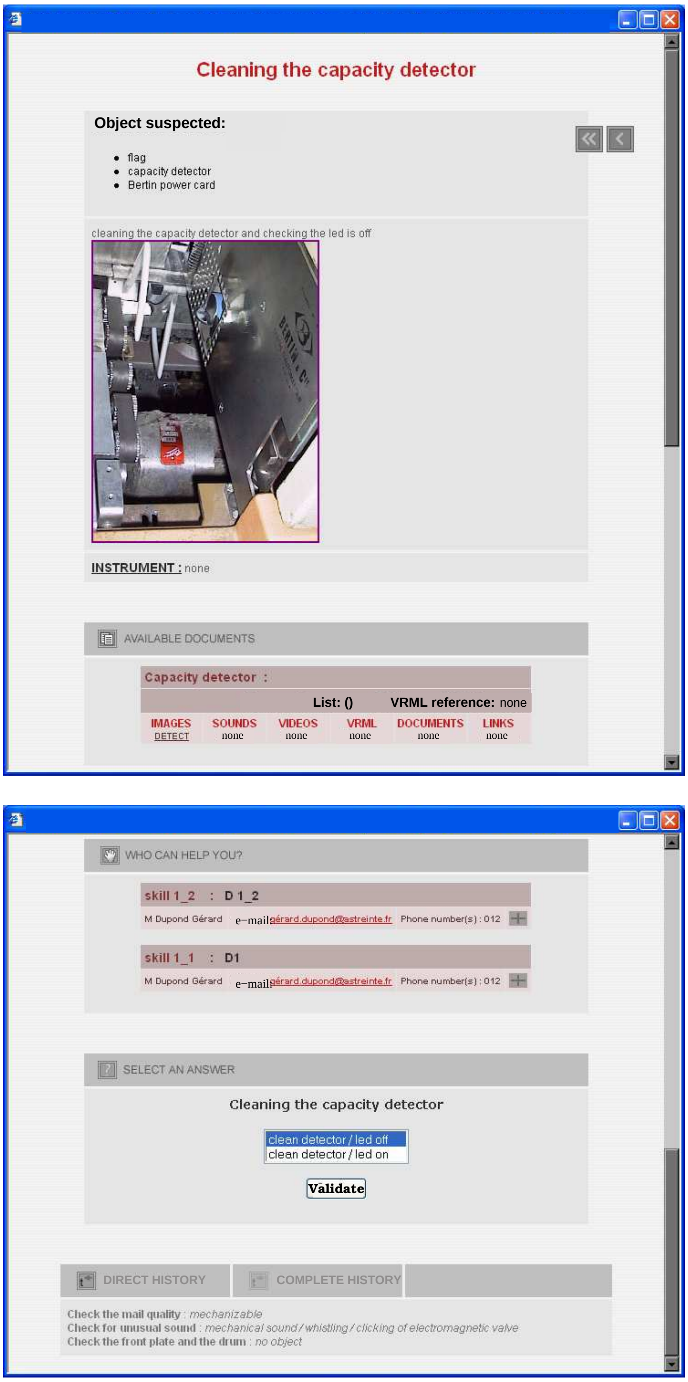


Figure 1: Example of a user’s task with Atanor.

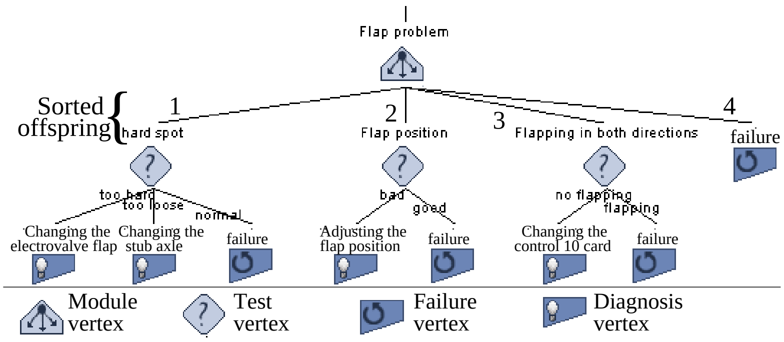


Figure 2: Formal representation of the knowledge by an *expert tree*.

through three modules: "Operator" for problem solving, "Expert" for knowledge acquisition and model updating, and "Manager" for analyzing the indicator panel.

Each model has its own graphical user interface and communicates with the Prolog kernel of the knowledge server.

3 Knowledge Visualization in Atanor

In the remaining of this paper, we focus on the "Expert" module which is the most complex. It allows to represent procedural knowledge associated with a know-how in different steps. For maintenance aid, a diagnosis with Atanor consists of several successive tests on the state of the components or on the functionnalities of the system from the easiest hypotheses to the most complex ones.

3.1 The "Expert Tree" Model

This graphical representation is a generalization of the decision trees and fault trees [Figure 2]. The vertices are classified into two categories: *(1)* the *module* vertices, not present in the classical decision and fault trees, are associated with a specific component of the system, and its offspring are organized from left (the easiest task) to right (the most difficult task). The order is predefined by the expert. Each of these vertices represents a step in the experts' strategy: the first son of a module allows to quickly find a decision with simple operations and the next sons are associated to tasks requiring more complex operations. *(2)* The *test* vertices are associated with a variable like in the decision trees; their incident arcs represent the different values of the variable associated to the states of the system components.

There are two types of leaves: those associated with a *diagnosis* which gives a solution to the problem detected, and those associated with a *failure.* In the latter, a backward procedure is launched.

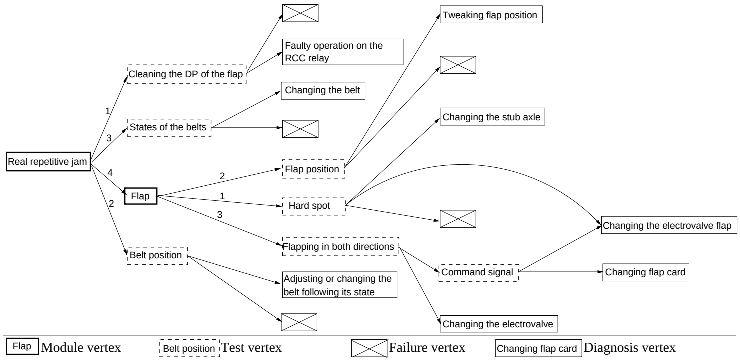


Figure 3: Example of a layered graph for the *expert* model.

An important property of this graphical representation is the transformation of the *expert* tree into a set of rules by a walk from the root to each leaves: "*If test_something then diagnosis"* or "*if test_something and another_test then diagnosis*". The operator "*and"* can be associated with a failure vertex or two consecutive *test* vertices.

The graphical representation is naturally more readable and synthetic than its equivalent set of production rules. Nevertheless, subtrees corresponding to rule subsets used at various stages can lead to many vertex redundancies causing reading difficulties and hiding important characteristics.

3.2 The Graph'Atanor Model

To overcome these difficulties, we developed the Graph'Atanor model based on layered digraphs. This model is a one-to-one representation of the internal Prolog model which associates a vertex with its offspring without any redundancy. With Graph'Atanor, like in the *expert* tree, the vertices represent the tests, modules, diagnoses and failures. The major difference is here uniqueness: a vertex cannot be duplicated. The vertices are sorted in vertical layers; layer i contains the vertices at distance i –following the shortest path on the drawing– from the initial layer. The arcs of the test vertices represent the different values of the associated variables. For the module, like in the *expert* tree, the arcs are associated according to an order which defines the priority of the transition [Figure 3].

The main goal of this representation is to provide a clear and intelligible layout. Although these notions call for subjective factors which closely depend on the audience, different criteria modeled by combinatorial constraints are commonly advocated by the information visualization community [Di-Battista et al., 1999]. When beside the physical constraints inherent to the medium (*e.g.* standard size sheet, computer screen), a drawing convention is given (here a layered drawing), aesthetics plays a fundamental role. Their optimization aims at facilitating both readability and memorization of the information embodied in the graph. Among the different possible criteria, recent experiments have confirmed that the minimization of arc crossings is by far the most important [Purchase, 2000]. Minimizing crossings in a layered layout could seem intuitively easier than the more general problem of minimizing crossings on the plane since the number of crossings is determined by the vertex ordering instead of the vertex geometric coordinates. Yet, it remains *NP*-complete even if there are only two layers [Garey and Johnson, 1983].

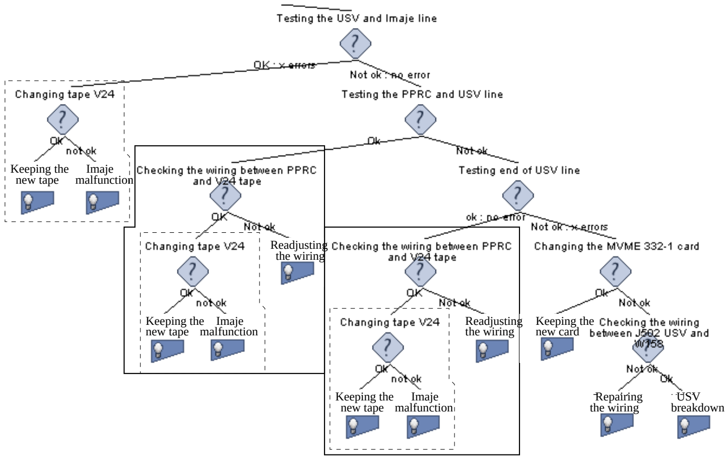
Numerous heuristics for this problem follow a layer-by-layer sweep scheme: the vertices of each layer are reordered to reduce crossings while holding the vertex orderings on the other layers. The most commonly used strategies for reordering are the sorting methods and the averaging heuristics. The sorting methods exchange vertices using crossing numbers in a way similar to classical sorts. The averaging heuristics are based on the idea that edge crossings tend to be minimized when connected vertices are placed facing each other. Recently, several metaheuristics have been developed for this problem like Tabu Search, GRASP and a hybridized genetic algorithm we developed [Kuntz et al., 2006]. The visual representation of Graph'Atanor is done with our hybridized genetic algorithm which follows the basic scheme of a genetic algorithm with two major differences: the use of two problem-based crossovers and a hybridization resulting from a local-search strategy. To ensure an intelligible graph layout, we relax the constraint which fixed an order on the offspring of a module vertex (a number above the arc shows the correct order).

4 Comparison of the Graphical Models

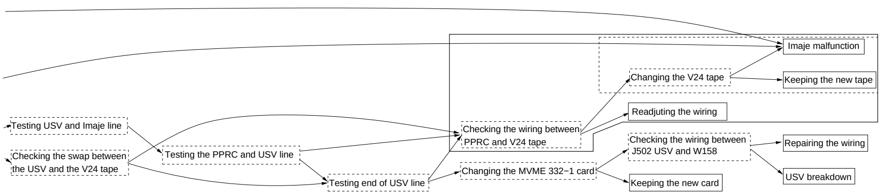
We have compared both models in the context of the maintenance aid for mail sorting machines (from the French company La Poste). After a training period on the *Expert* module, four geographically dispersed experts of the machine built and then updated the knowledge model carried by the server. This process took about two years. The experts listed thirty different possible breakdowns with an *expert* tree for each one. The trees contain more than 400 test vertices and 200 different diagnoses.

[Figure 4-a] is a part of an *expert* tree for repairing a specific breakdown. [Figure 4-b] is a part of the Graph'Atanor model for the same breakdown.

The graph-based representation has several advantages. We can easily see that an identical diagnosis can be reached quickly in few steps, or slowly through many tests. This information is important for a manager. In addition, some tests (*e.g.* the tests in the first layer) can be considered critical because a correct answer is required to avoid unnecessary operations. Statistically, the vertices with a high degree are used more frequently by the operators and Graph'Atanor allows to analyze the various walks of the graph. This is useful to improve the planning of the preventive maintenance tasks of the components associated with highly engaged vertices. These require much attention as they could indicate a weakness in the system.



(a) The *expert* tree. The surrounded subtrees show some identical parts of the layout.



(b) The graph layout. The surrounded subgraphs show the duplicated parts of the *expert* tree.

Figure 4: Part of a job process for the maintenance aid for the mail sorting machines.

5 Conclusion

Including well-adapted visualization supports in KMS can significantly improve the interactivity with the users and consequently make particular knowledge collecting and information retrieval more efficient. In this paper we have developed a graphical framework for the expertise phase with the KMS Atanor. The model based on layered digraphs has proved more suitable than the more classical tree-based models.

Extensions are currently under study for increasing the interactivity of the displays. It is true that, the construction of the ruleset is intrinsically dynamical, consequently the visual representations should be adapted to take into account a series of modifications. More formally, at each step t of the process, a graph G_t is proposed. G_t must satisfy two requirements: it has to remain readable and visually as close as possible to the drawing at step $t-1$ to preserve the user's mental map [Eades et al., 1991]. This problem can be set as a multiobjective optimization problem which is still open, as far as we know [Branke, 2001]. We have recently proposed an extension of the genetic algorithm presented here for the dynamical graph drawing problem [Pinaud et al., 2004]. The results have shown to be promising on graph series issued from an automatic generator. We plan to integrate this approach into a new version of the KMS Atanor.

References

[Branke, 2001] Branke, J. (2001). Dynamic graph drawing. In Kaufmann, M. and Wagner, D., editors, *Drawing Graphs: methods and models*, pages 228–246. Springer.

[Burkhard, 2004] Burkhard, R. (2004). Learning from architects: The difference between knowledge visualization and information visualization. In *Proc. of the* 8^{th} *Int. conf. on Information Visualisation (IV)*, pages 519–524.

[Buzan and Buzan, 1996] Buzan, T. and Buzan, B. (1996). *The Mind Map Book*. Plume Books.

[Corby and Dieng, 1998] Corby, O. and Dieng, R. (1998). Webcokace: a commonKADS expertise model web server. In *Proc. of the* 11^{th} *Workshop on Knowledge acquisition, Modeling and Management (KAW)*.

[Di-Battista et al., 1999] Di-Battista, G., Eades, P., Tamassia, R., and Tollis, I.-G. (1999). *Graph drawing – Algorithms for the visualization of graphs*. Prentice-Hall.

[Eades et al., 1991] Eades, P., Lai, W., Misue, K., and Sugiyama, K. (1991). Preserving the mental map of a diagram. In *Proc. of Compugraphics*, pages 24–33.

[Eppler and Burkhard, 2005] Eppler, M. and Burkhard, R. (2005). Knowledge visualization. In Schwartz, D., editor, *Encyclopedia of Knowledge Management*. Idea Group.

[Garey and Johnson, 1983] Garey, M. and Johnson, D. (1983). Crossing number is NP-complete. *J. Algebraic Discrete Methods*, 4(3):312–316.

[Guillet et al., 2002] Guillet, F., Follut, D., Philippé, V., and Philippé, J. (2002). Athanor: Une approche pour la gestion des connaissances de maintenance sur les systèmes complexes. In 1^{ere} *journée SI pour l'aide à la décision en ingénierie système (JESIADIS)*, pages 41–54.

[Herman et al., 2000] Herman, I., Melançon, G., and Marshall, M. (2000). Graph visualization and navigation in information visualization: a survey. *IEEE Trans. on Visualization and Computer Graphics*, 6(11):24–43.

[Kuntz et al., 2006] Kuntz, P., Pinaud, B., and Lehn, R. (2006). Minimizing crossings in hierarchical digraphs with a hybridized genetic algorithm. *J. of heuristics*, 12(1–2):23–36.

[Lehmann, 1992] Lehmann, F. (1992). Semantic network. In Lehmann, F., editor, *Semantic Networks in Artificial Intelligence*, volume 24 of *Int. Series in Modern Applied Mathematics and Computer Science*, pages 1–50. Pergamon Press.

[Limnios, 2005] Limnios, N. (2005). *Arbres de défaillances (*2^e *Ed.)*. Hermes.

[Pinaud et al., 2004] Pinaud, B., Kuntz, P., and Lehn, R. (2004). Dynamic graph drawing with a hybridized genetic algorithm. In *ACDM VI*, pages 365–375. Springer.

[Purchase, 2000] Purchase, H. (2000). Effective information visualisation: a study of graph drawing aesthetics and algorithms. *Interacting with computers*, 13(2):147–162.

[Quilan, 1990] Quilan, J. (1990). Decision trees and decision making. In *IEEE SMC*, number 20, pages 339–346.

[Roche et al., 2005] Roche, C., Foveau, C., and Reguigui, S. (2005). La démarche ontologique pour la gestion des compétences et des connaissances. In *Proc. of EGC*, pages 359–364. Cépaduès éditions.

[Sowa, 1992] Sowa, J. (1992). Conceptuals graphs summary. In Eklund, P., Nagle, T., Nagle, J., and Gerholz, L., editors, *Conceptual Structures: Current Research and Practice*, pages 3–52.

[Vergnaud et al., 2004] Vergnaud, N., Harzallah, M., and Briand, H. (2004). Modèle de gestion intégrée des compétences et connaissances. In *Actes Conf. Extraction et Gestion des Connaissances (EGC)*, pages 159–170. Cépaduès Editions.