Visualizing Dependencies Between Digital Product Artefacts - Creating a Visualization Layout Based on a User Study

Grischa Beier, Robert Müller
Department Model-based Engineering
Fraunhofer IPK
Berlin, Germany

***Abstract*— Along the development process of a product numerous digital artefacts are created. The explicit visualization of the dependencies between these artefacts supports developers in comprehending their product. This paper presents a novel visualization layout for compound graphs displaying product dependencies, which is based on the findings of a user study and is implemented in a software prototype.**

***Keywords*— Traceability Visualization; Graph Layout**

I. INTRODUCTION

During a product development process numerous documents or models are created [1]. Each of these digital artefacts (such as a requirements specification or a product structure) describes the to-be-developed product from a different, increasingly refined perspective [2]. Traceability is a development method where dependencies between elements of these artefacts are explicitly modelled [3]. One of the main advantages of traceability is, that it supports developers in comprehending their product more easily [4, 5] especially when visually displaying the dependencies – which are also referred to as Tracelinks [6].

However, an adequate visualization for traceability information has not been investigated sufficiently yet [7]. Therefore the paper aims at systematically developing such a visual representation. The following section provides a state of the art analysis regarding traceability visualization. It is followed by a user study comparing different graph layouts, then a subsequent advancement of the best layout which is succeeded by an introduction of a prototypical visualization software.

II. TRACEABILITY VISUALIZATION

There are three established forms to visually represent traceability information: a matrix (especially Design Structure Matrizes), node-link-diagrams and lists (with references). Often software tools offer more than one form of visualization: IBM Rational Doors® (all three), LOOMEO® (matrix and node-link), Dassault V6® (node-link-diagram and lists), Siemens Teamcenter® 9.1 (matrix and lists) etc. But none of the existing tools satisfies all of the major visualization requirements that are important when developing technical systems:

- allow for a flexible number and different types of artefacts,
- allow for a display of the hierarchical structure of artefacts,
- allow for hierarchical transitivity of artefacts (e.g. show tracelink on parent element, when child element is hidden).

To date there are three studies comparing the mentioned representation forms in different usage scenarios [8, 9 and 10]. The main findings of [8] suggest, that the choice for an adequate visual representation for complex artefacts depends on the tasks. For the two tasks where users had to deal with the actual content of the graph the node-link-representation was better or at least equal to the matrix representation. The study presented in [9] was based on a graph representing an actual product model - though non-hierarchical, too. It was found that node-link is favourable compared to matrix representation if the graph represents a semantic context. In study [10] different kinds of tasks were investigated. They discovered that generally node-link and matrix are perceived as better readable than list representations. Regarding the design task the node-link and regarding the management task the matrix representation were evaluated as having the best information value. All three studies do not investigate compound graphs comprising multiple hierarchical artefacts which is limiting their significance for technical Systems Engineering.

For that reason the feedback from industrial applicants presented in [6] is especially valuable. They found the presented matrix solution “somewhat daunting” and the displayed content difficult to absorb. These impressions correlate with the results of a not-yet-published user study by the authors, where participants also favoured node-link over matrix and list representations.

III. NODE-LINK LAYOUT ARIADNE’S EYE

Due to the mentioned reasons it seems likely that a node-link representation is the most promising visualization for hierarchical compound graphs, which is why the authors decided to elaborate the node-link layout Ariadne’s Eye, that was also inspired by the Hierarchical Edge Bundles presented in [11] and [12]. Ariadne’s Eye follows the above mentioned major visualization requirements by

- spatially separating artefact information (incl. hierarchical relations) from tracelinks by positioning the artefacts outside a common circle through which the artefact-spanning tracelinks run,
- dividing the circle into as many sections as artefacts need to be displayed (see Figure 1 - left); allowing for a flexible

number of artefacts: by showing one artefact per section and

- positioning elements from the same hierarchical level on a common circular layer to easily perceive the hierarchical order of elements (see Fig. 1 - right).

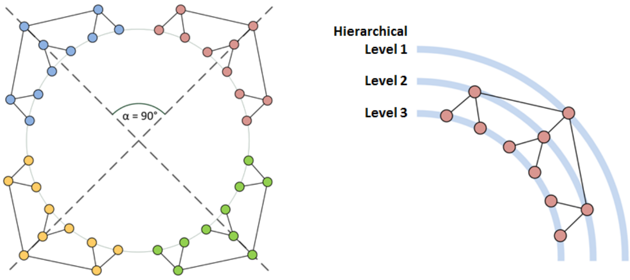

Fig. 1. Basic principles of Ariadne's Eye: Circle Segmentation per artefact (left) and positioning of elements according to their hierarchical level (right)

IV. USER STUDY

To verify the hypothesis that a spatial separation of artefact and tracelink information leads to an improved readability (thereby also verifying the suitability of the developed layout Ariadne's Eye) the authors performed a user study in which Ariadne's Eye (L3) was compared to two establishes node-link layouts (a vertical tree layout (L1) and a balloon tree layout (L2) - see Appendix). All three graphs were displaying the requirements, functions and product structure of a hypothetic air conditioning system: three artefacts with in total 97 labeled elements and 179 tracelinks between them.

A. *Study design*

21 participants took part in the study. The independent variables were graph layout and task, while the dependent variables measured were the correctness of the identified elements, the time required to identify them (as suggested in [8] and [13] to measure the readability in paper and pen studies) and the subjective preference of the participants.

Every participant was asked to deal with three tasks[1]. The tasks were adapted from classical readability tasks in order to suit hierarchical compound graphs, making sure users can identify an element in its hierarchical context:

Task 1: Please find and mark the requirement "Noise-free continuous operation". Which third-level functions F 1.x.x are directly linked to the requirement "Noise-free continuous operation"? Please mark and note their identifier!

The parent nodes of these identified third-level functions are second-level functions F 1.x. Mark and note the identifiers of all elements in the product structure that are directly linked to the identified second-level functions F 1.x!

Task 2: Please find and mark the requirement "Legal requirements".
Which elements in the product structure are directly linked to the requirement "Legal requirements"? Please mark and note their identifier!

Which functions are directly linked to the requirement "Legal requirements"? Please mark and note their identifier!

Task 3: *Please find and mark the function "Cool down air" and all its child nodes.*

How many direct links does the function "Cool down air" have with the

- *product structure,*
- *requirements?*

How many direct links do all child nodes of the function "Cool down air" have with the

- *product structure,*
- *requirements?*

All participants worked on the three tasks in the same order while the graph layout provided for each task was alternated allowing for an equal distribution.

B. *Results & conclusion*

The quantitative analysis of the study for the dependent variables number of errors and time required showed the results displayed in **Error! Reference source not found.**, where the mean values per task and layout are given.

TABLE 1. MEAN VALUES FOR THE DEPENDENT VARIABLES NUMBER OF ERRORS AND TIME REQUIRED PER TASK AND LAYOUT

		Task 1		Task 2		Task 3	
		Errors	Time [s]	Errors	Time [s]	Errors	Time [s]
Layout	L1	0,86	358,71	2,43	498,00	6,14	295,00
	L2	1,43	334,71	1,71	353,29	5,00	285,57
	L3	0,29	363,86	1,00	429,29	5,00	166,71

The results show that working with Ariadne's Eye (layout L3) leads on average to the least number of errors. Regarding the time required to solve the tasks no single layout seems to be superior. To investigate whether the layout has a significant impact on either of the two dependent variables an analysis of variance (ANOVA) has been performed. The values for the probability (p-value; threshold 5%) of the correctness of the null hypothesis2 per task are given in **Error! Reference source not found.**.

TABLE 2. ANOVA RESULTS FOR ERRORS AND TIME PER TASK

	p (Task 1)	p (Task 2)	p (Task 3)
Errors	0,157	0,423	0,956
Time	0,900	0,187	0,042

[1] The entire study was performed in German language. Therefore the three tasks recited here are an English translation of the German original.

[2] The chosen null hypothesis was: the type of layout does not influence the dependent variables

The ANOVA results show, that null hypothesis can at least be rejected for the time required to solve task 3. This means that the type of layout does have a significant impact at least on the dependent variable required solution time - in favour of layout L3: the newly developed Ariadne's Eye. For the other two tasks no significance regarding the solution time could be detected. The same applies for the errors - although least of them were committed with Ariadne's Eye in all tasks - it could not be proven that the graph layout has a significant impact on the error rate. These uncertainties should be investigated in more detail in future user studies with a higher number of participants also analysing the parameters for the respective subtasks.

On the other hand, when asked which of the given three layouts they preferred, 67% of all participants voted for Ariadne's Eye (L1: 14%; L2: 19%). These two facts, the subjective preference as well as the objective and significant superiority for one task, lead to the decision to further develop the layout Ariadne's Eye towards a visualization tool for compound graphs.

V. ADVANCING THE LAYOUT ARIADNE'S EYE

The user study revealed some deficiencies of the layout. Apart from the generally positive feedback some participants commented that the rather big distance between some elements of different artefacts was a flaw, while others were suspecting that the drawing area between the artefacts was not efficiently used. This feedback was taken as a motivation to further improve the layout.

Therefore a mathematical analysis was undertaken to investigate how efficiently the traceability information can be visualized comparing three geometrical forms (circle, eclipse, polygon) each with three and four artefacts respectively (n = number of artefacts). The following parameters were used to determine the efficiency:

- number of label overlaps,
- average tracelink length[3],
- overflow of visualization objects (node or label) and
- percentage of used drawing area.

To ensure comparability between the layouts all nodes and labels had the same size, all artefacts had the same number of hierarchical levels (7) and elements per level (1, 10, 20, 30, 40, 50, 60) and the distance between neighbouring same-level-nodes was normed to 1. All labels are oriented horizontally to allow for an easier readability (see Fig. 2).

[3] Assumptions: 1) only the nodes of the lowest hierarchical level are considered; 2) every node is linked to all other nodes of the other artefacts

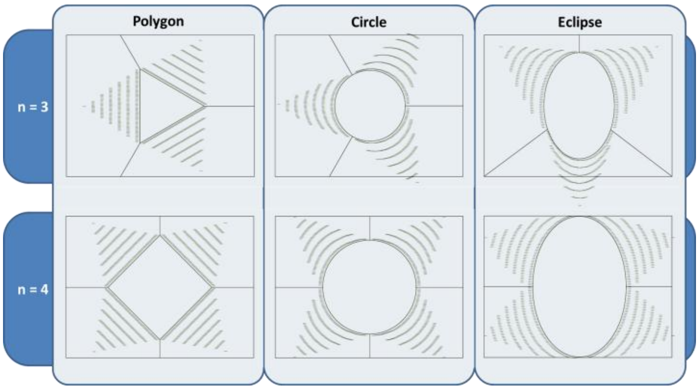

Fig. 2. Comparison between polygon, circle and eclipse

Error! Reference source not found. shows the results of the analysis:

TABLE 3 ANALYSIS' RESULTS FOR THE MOST EFFICIENT GEOMETRICAL FORM TO VISUALIZE TRACEABILITY COMPOUND GRAPHS

	Circle		Polygon		Eclipse	
	n = 3	n = 4	n = 3	n = 4	n = 3	n = 4
Label overlaps	22,9%	10,3%	0%	0%	12,8%	2,8
Tracelink length	44,7	60,3	36,9	53,2	60,2	75,7
Object overflow	1,7%	0%	0 %	0 %	9,6%	0,2%
Used area	51%	72%	46%	68%	58%	88%

The polygon-form outperforms the other forms in three out of four categories. Only in the field of used drawing area the eclipse achieves better results. For that reason and since the readability of labels is more crucial than information density when it comes to interpreting graphs the polygon is chosen as the most suitable geometric form for displaying compound graphs. Therefore the layout Ariadne's Eye is adapted so that the geometric foundation along which the artefacts are positioned will be a polygon.

Furthermore Ariadne's Eye was advanced by an interaction concept (scaling, zooming, folding etc.) that helps users to grasp a selected element, its hierarchical context as well as all linked elements and their direct ancestral line more easily. The developed visualization tool Ariadne's Eye is displayed in Fig. 3 (see a bigger version in Fig. 7 as part of the Appendix).

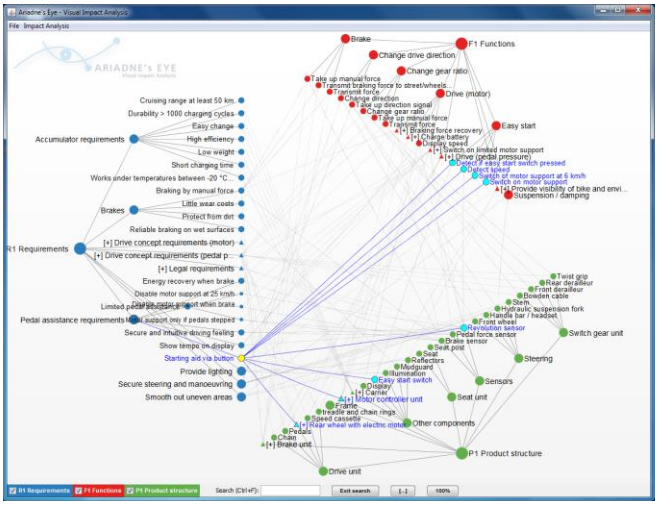

Fig. 3. Traceability visualization prototype Ariadne's Eye

VI. SUMMARY

This paper presented the systematic development and evaluation of an innovative visualization layout for compound graphs (which need to relate multiple hierarchical artefacts) as required in the context of traceability visualization. By the means of a user study it was demonstrated that the spatial separation of artefact and tracelink information leads to an improved readability of such hierarchical graphs. The validity of the study is mainly limited by two factors though: the relatively low number of participants and the quantity of elements within an artefact that is far from industrial artefact dimensions.

Based on the initial feedback from the study the layout was further refined and advanced. The final concept was implemented as an interactive visualization tool called Ariadne’s Eye – for further detail information on the tool Ariadne’s Eye see [14] and [15]. The introduced layout Ariadne’s Eye is a significant contribution to the research on traceability visualization and generated very positive initial feedback when demonstrated during industry workshops.

REFERENCES

[1] Beier, G.; Figge, A.; Marwedel, S. (2017): A MBSE approach for the development of complex technical systems. In: International Journal of Engineering Research and Applications, (7) 9, S. 24-36.

[2] Beier, G.; Rothenburg, U.; Woll, R.; Stark, R. (2012): Durchgängige Entwicklung mit erlebbaren Prototypen - Modellbasiertes Systems Engineering. In: Digital Engineering, 3/2012, 14-17.

[3] IEEE (Institute of Electrical and Electronics Engineers) Computer Society. 1990. *Standard Glossary of Software Engineering Terminology*. New York, USA: IEEE Computer Society Press.

[4] Marcus, A., Xie, X., and Poshyvanyk, D. 2005. “When and How to Visualize Traceability Links?” In *Proceedings of the Third International Workshop on Traceability in Emerging Forms of Software Engineering*, 56-61. New York, USA: ACM Press.

[5] Chen, X. 2010. “Extraction and Visualization of Traceability Relationships between Documents and Source Code.” In *Proceedings of the IEEE/ACM international conference on Automated software engineering*, 505-510. New York, USA: ACM Press.

[6] Jarratt, T., Eckert, C., and Clarkson, P. J. 2004. “Development of a Product Model to Support Engineering Change Management”. In *Proceedings of the Fifth International Symposium on Tools and Methods of Competitive Engineering*, edited by L. Horvath and P. Xirouchakis, 1-12. Rotterdam, Netherlands: Millpress.

[7] Winkler, S. and Pilgrim, J. 2010. “A Survey of Traceability in Requirements Engineering and Model-Driven Development.” *Software & Systems Modeling* 9 (4): 529-565. doi: 10.1007/s10270-009-0145-0

[8] Ghoniem, M., Fekete, J.-D., and Castagliola, P. 2004. “A Comparison of the Readability of Graphs Using Node-Link and Matrix-Based Representations.” In *Proceedings of the Symposium on Information Visualization*, Vol. 4, 17-24. Washington, USA: IEEE Computer Society Press.

[9] Keller, R., Eckert, C. M., and Clarkson, P. J. 2006. “Matrices or Node-Link Diagrams: Which Visual Representation is Better for Visualising Connectivity Models?” *Information Visualization* 5 (1): 62–76. doi: 10.1057/palgrave.ivs.9500116

[10] Li, Y.and Maalej, W. 2012. “Which Traceability Visualization Is Suitable in This Context? - A Comparative Study.” *Requirements Engineering: Foundation for Software Quality. Lecture Notes in Computer Science* (7195): 194–210. doi: 10.1007/978-3-642-28714-5_17

[11] Holten, D. (2006): Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data. In: IEEE Transactions on Visualization and Computer Graphics, 12 (5), S. 741–748.

[12] Holten, D.; Cornelissen, B.; Van Wijk, J. J. (2007): Trace Visualization Using Hier-archical Edge Bundles and Massive Sequence Views. In: Proceedings of the 4th IEEE International Workshop on Visualizing Software for Understanding and Analysis, VisSoft '07, Alberta, Kanada, Juni 24-25, IEEE Computer Society Press: Los Alamitos, USA, S. 47–54.

[13] Purchase, H., Cohen, R., and James, M. 1997. “An Experimental Study of the Basis for Graph Drawing Algorithms.” *The ACM Journal of Experimental Algorithmic* 2 (1). doi: 10.1145/264216.264222

[14] Beier, G. (2014): Verwendung von Traceability-Modellen zur Unterstützung der Entwicklung technischer Systeme. Dissertation, Technische Universität Berlin, Fakultät für Verkehrs- und Maschinensysteme, Fraunhofer-Verlag.

[15] Brandenburg, E.; Figge, A.; Zander, S.; Beier, G. (2014): Recommendations for Tracelink Decisions – An Empirical Investigation of Visualization Methods. In: Proceedings of the 5th International Conference on Applied Human Factors and Ergonomics, Krakow, Poland.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 12, December - 2017

APPENDIX

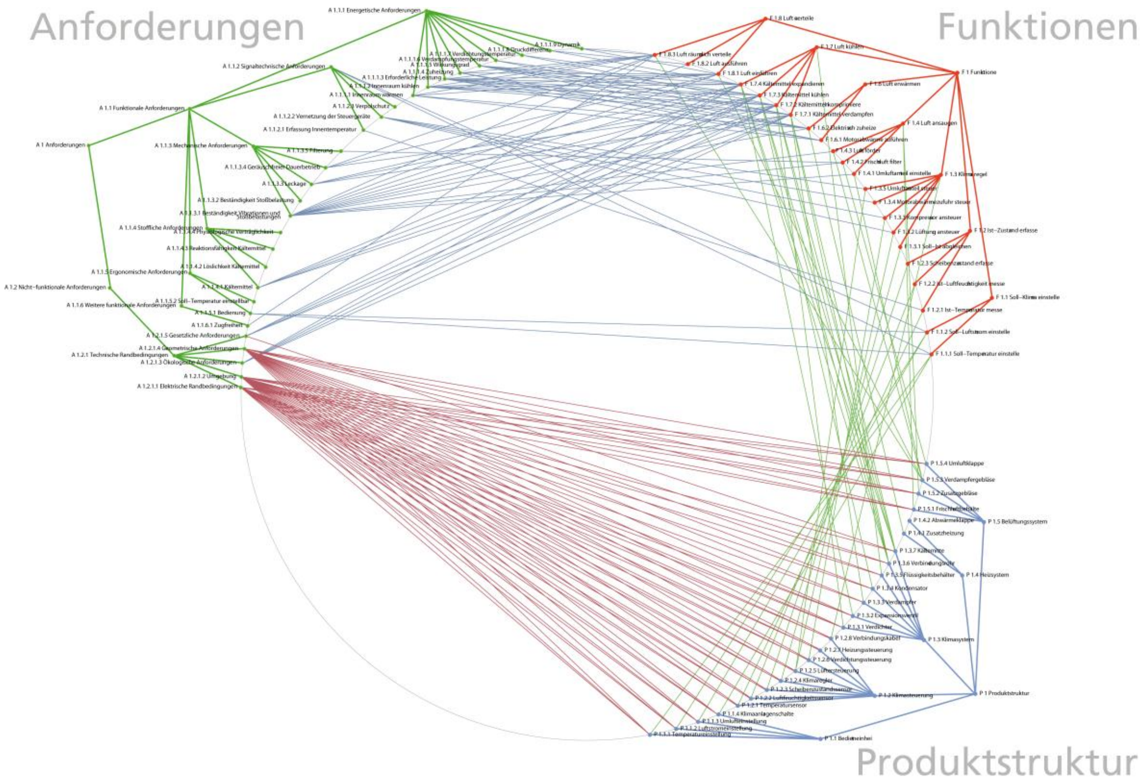

Fig. 4. Ariadne's Eye Layout

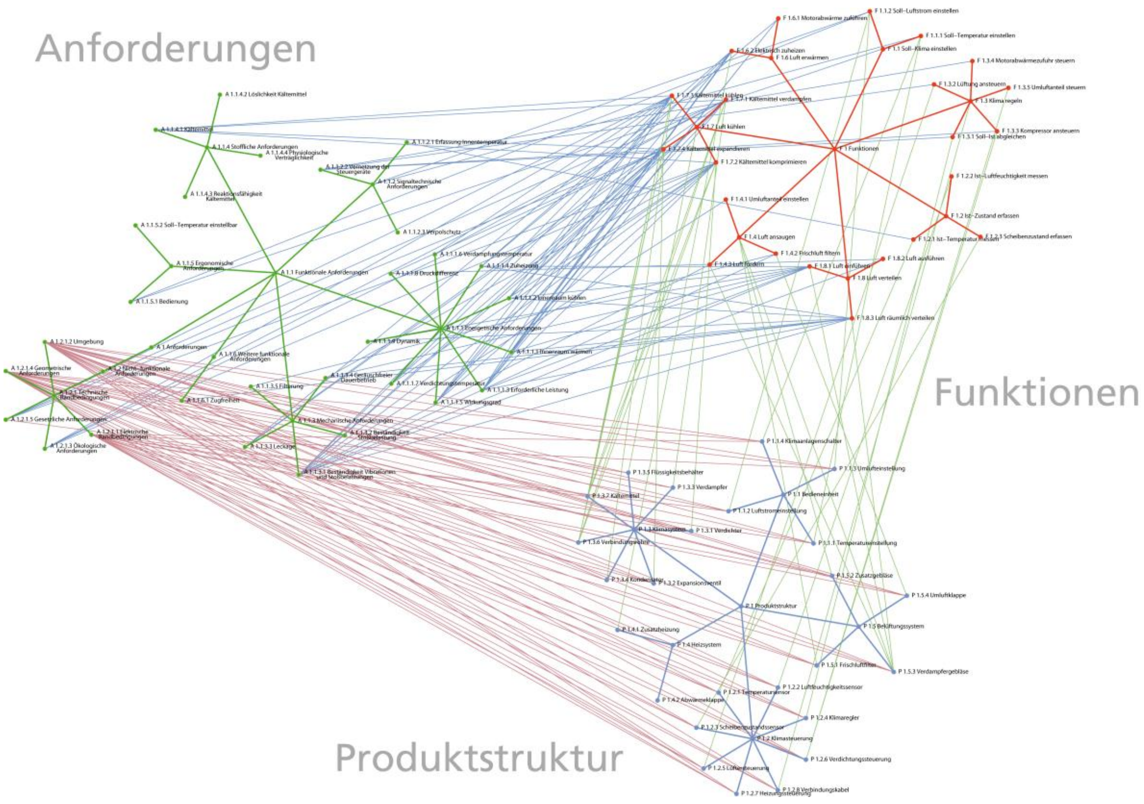

Fig. 5. Balloon tree layout

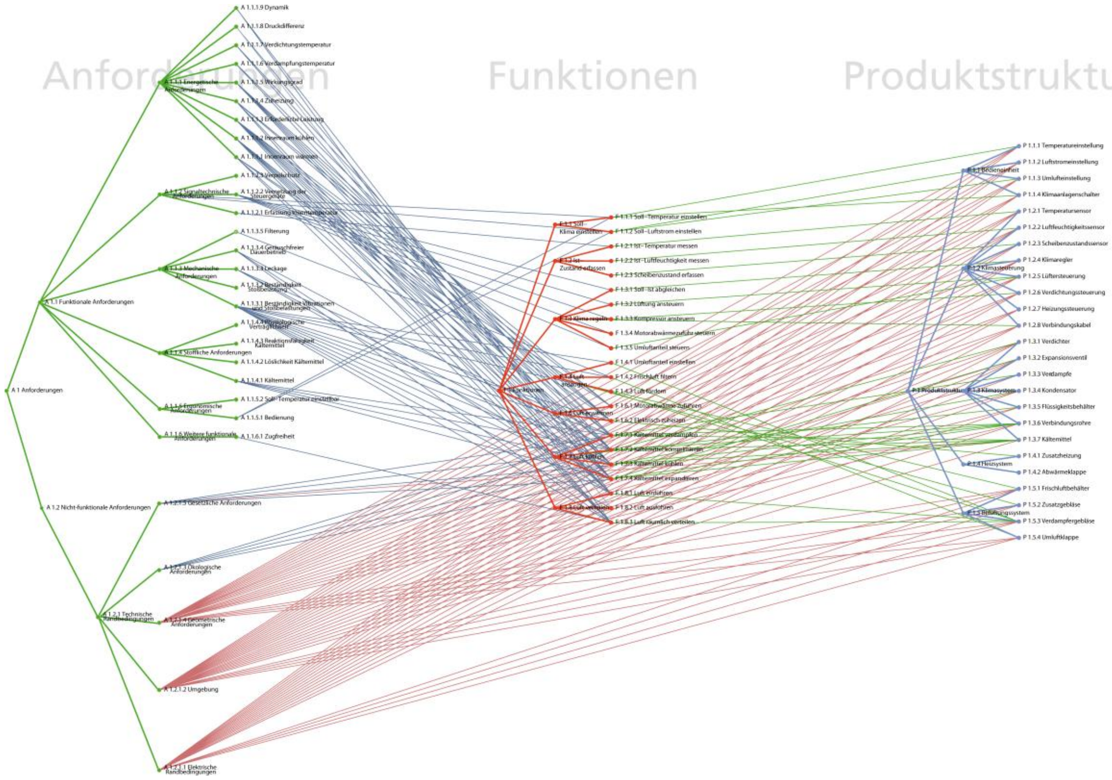

Fig. 6. Vertical tree layout

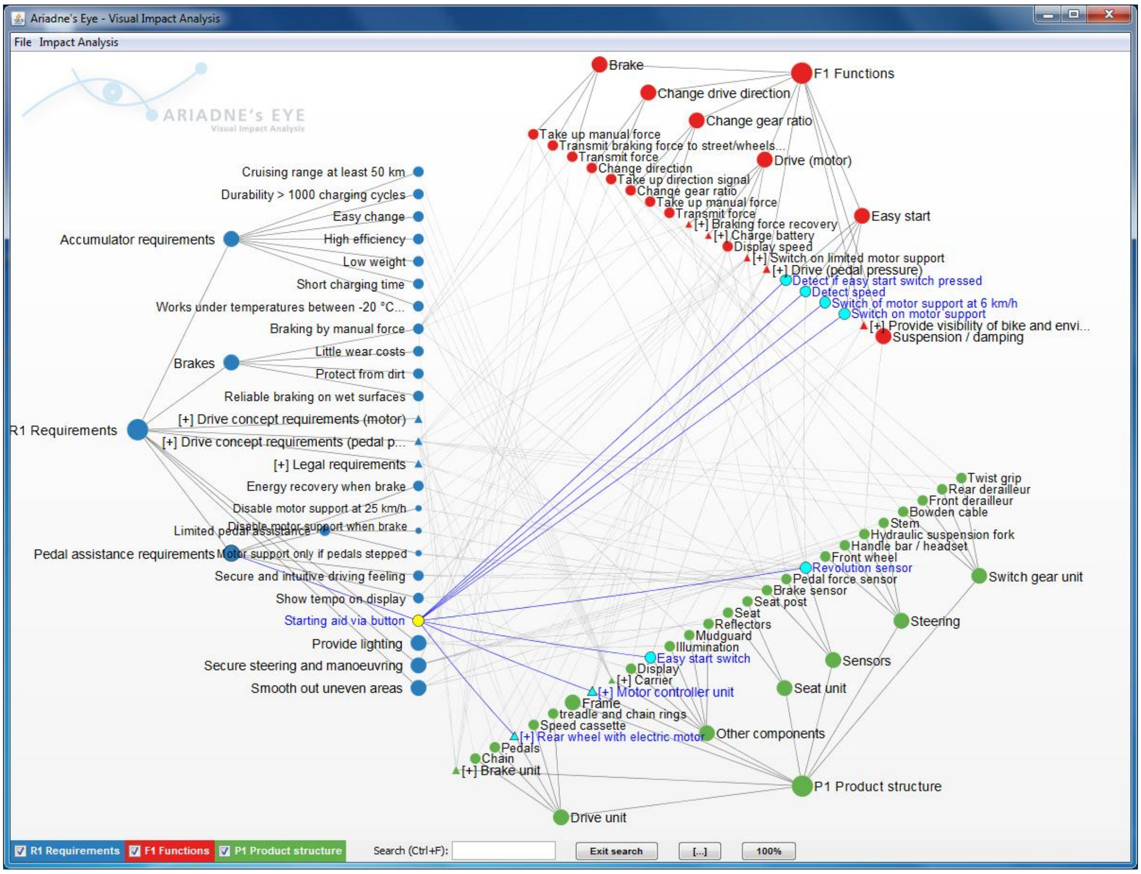

Fig. 7. Traceability visualization prototype Ariadne's Eye (bigger version)