The Confluence Diagram: Embedding Knowledge in Interaction Constraints

Martin J. Eppler* — University of St. Gallen

Sebastian Kernbach† — University of St. Gallen, University of Lugano

Benjamin Wiederkehr‡ — Interactive Things

Peter Gassner∮ — Interactive Things

Abstract

In this paper we propose an interactive knowledge visualization application that enables team members to integrate their knowledge about a project's parameters (such as budget, scope, staffing etc.) through a visual, interactive configurator that also captures dependencies among these factors visually. The novelty of our approach can be seen in the fact that the diagram encodes knowledge from the participants in the interaction constraints encoded by and later on imposed on the user by the diagram. The dependencies among key factors of the team are encoded as sliders that move together. These dependencies help the team members understand and align their action constraints and allow them to record their experiences for subsequent teams in the diagram, as well as track their on-going project constellation. First focus groups with typical users of the confluence diagram (project managers) show that this is a highly useful and useable property that other diagrams cannot offer as of today.

Keywords: Diagrams, Knowledge Visualization, Experience Sharing

1. Introduction

Amidst the on-going, justified hype and thus research and practice focus on visual analytics and big data, there is a risk to neglect (for lack of a better term) big knowledge – the wealth of experience and insights in the form of (mostly) implicit knowledge of employees.

The key idea behind the emerging discipline of knowledge visualization is consequently to visualize not only mass data for improved decision making, but to also make the long-standing experience of experts and decision makers tangible in order to improve decisions, plans, and actions in organizations.

This paper reports an on-going project that aims at visualizing the – often implicit, unarticulated and thus undocumented – experiences, opinions, judgments, and insights of key stakeholders in a project. As a first step, we have built an interactive diagrammatic visualization that helps teams integrate their members' knowledge about key parameters and secure their collective wisdom about project constraints and dependencies. The basic premise of this approach is that we will not be able to make all decision-relevant knowledge machine-readable or accessible through semantic technologies. It is thus important to give users tools through which they can make their own implicit knowledge explicit, combine it with each other, and store the resulting knowledge integration results in an accessible digital format that lets others re-use their insights productively. This is the aim of the confluence diagram application.

2. Rationale and Overview

Many collaborative endeavors rely on diverse team members' specialized expertise [1]. This expertise regarding the project's constraints and dependencies should be made available to all team members in a joint view that provides overview (the project's overall shape) and detail on demand (individual parameters of the project).

The confluence diagram uses the radar [2, 3], spider, or cobweb chart for this purpose and combines it with dynamic, interactive sliders to capture the team members' views. In the following use case we illustrate this idea through a real-life example. We then outline the technical architecture as well as usability concept of the tool. We conclude by describing on-going development efforts.

3. Use Case

An IT project team gathers for the kick-off meeting. They use an empty confluence diagram projected on a large screen to build common ground and make sense of their situation. They (with the help of a facilitator) first create a small circle in the center and label it as "Intranet Re-launch". Then they segment the circle outside the small one into four ***dimensions*** labeled as *scope*, *resources*, *timing*, and *environment* that they believe determine (and constrain) their project work. Next, they define important ***factors*** in each of these four dimensions *(Fig. 1)*. For each factor they define a ***spectrum***.

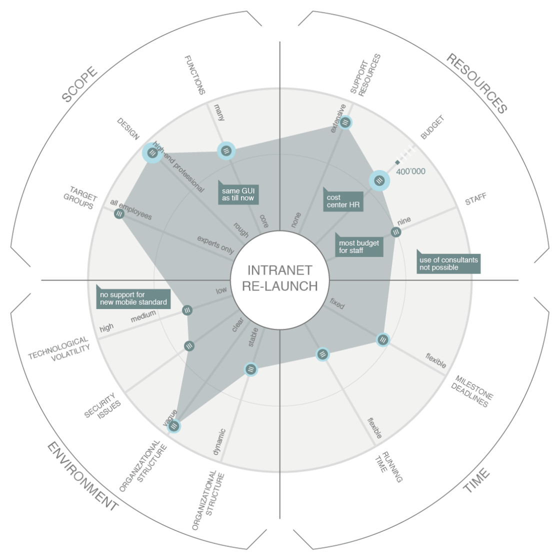

Figure 1: The confluence diagram to configure the parameters of a project

After all the spectrums have been defined, the project team members define where their project is located on the spectrum. They move a ***slider knob*** on each spectrum to a position that represents their perceived situation. In Fig. 1 they agree that the project's budget has been fixed at 400'000 with the possibility of additional 50'000 Euros if needed (represented as a barrier above the knob). This step will yield a status-quo configuration of the project and a corresponding *profile* of their ***solution space*** (the profile line/area connecting all slider knobs). This way the team will have a first visual insight into to the constraining qualities of their project.

As a next step, the team members discuss each factor's ***impact*** on other factors in the same or other dimensions. If, for example, the target groups of the intranet are reduced, this will also allow the team to reduce the budget and it will also affect the usability requirements of the Intranet. To define these dependencies the facilitator simply

* e-mail: martin.eppler@unisg.ch
† e-mail: sebastian.kernbach@unisg.ch
‡ e-mail: benjamin@interactivethings.com
∮ e-mail: peter@interactivethings.com

clicks on the affected knobs and on a “+”sign (for a ‘more leads to more’ relation) or “-“ sign (for ‘more leads to less’). The more a factor affects other factors, the larger the slider knob’s halo automatically becomes. Whenever the facilitator moves the mouse over a certain slider knob, all affected knobs will automatically be highlighted. When that slider knob is moved, all affected knobs will also move in the defined direction. Based on these relationships, the team now examines the ramifications of different changes by moving a few of the larger slider knobs. The team can discuss various scenarios for the project (such as a sudden budget reduction or a reduction in the project’s running time).

In a final step the project team uses the inner space of the profile line to place ***keywords***/fixed parameters for each factor. It places ***out-of-scope features*** outside the profile area. This is a case of data-aware annotation as described by [5]. The team ends the meeting with a much clearer, and more aligned understanding of the project’s scope, resource situation, timing, and environmental situation. The team members understand the dependencies among key factors of the project much better and know what monitor more closely. In addition, their insights into which factors matter and how they influence one another is stored for later project teams that can benefit from these experiences by using and adapting the pre-defined confluence configurator.

4. TECHNICAL ARCHITECTURE

The technical architecture is planned around the needs of the users: (I) embedded into users’ contexts, collaboration; (II) direct manipulation, immediate feedback; (III) worry-free exploration, representation and communication of solution spaces over time.

To address (I) a single-page web application communicates with a server component through a REST API which does not require installation, allows offline-usage through Local Storage, and can be embedded in PowerPoint presentations using the LiveWeb Add-In.

To achieve the fluid rendering and interaction performance required for (II), an immediate-mode GUI framework based on the React and Mori JavaScript libraries [6] is used.

For (III) immutable data-structures enable an implementation of solid undo/redo functionality, version snapshot, offline-use and auto-save functionality because all state can be easily (de-) serialized [7].

5. USABILITY CONCEPT

The usability concept of the diagram is based on the following key principles:

1. Free ride [8]: using the diagram generates an “at one glance” insight when looking at the resulting profile. The radial plot enables the users to detect whether the parameters in a dimension are homogenous or differ widely. It also allows detecting areas where constraints are higher than in others. In addition, the characteristic shape of the resulting profile line lets users efficiently compare different project constellations.

2. Affordance [9]: the knobs invite changes and modifications to the scales and provide an incentive to try out alternative scenarios and examine their effects on other areas of the diagram

3. Fluid-rigid state representation: once the configurator has been used to find an adequate situational representation, the team members can decide to freeze the image and signal its finishedness to others by changing its appearance from a provisional graphic to a more polished one.

6. EVALUATION AND ON-GOING DEVELOPMENTS

In order to evaluate the application at this stage, we have conducted focus groups with a total of 25 future users. These experienced managers assessed the strengths, weaknesses, and development vectors of the interactive application, after having been given a detailed explanation of the diagram’s functioning and application. Regarding strengths, the managers voiced that the diagram would be a great tool to teach project management and to communicate a complex project to the steering committee. It would help build a common understanding in a project team and it is instructive to compare typical project constellations in project evaluations. With regard to weaknesses they saw a risk in the diagram to oversimplify dependencies and give a false sense of relations (quasi-linear) if expectations are not properly managed. In terms of development opportunities the participants saw great potential in combining this tool with typical system dynamics software packages and building up a library of useful templates for the tool. They also agreed that adding a second, coordinated screen would make sense, in order to enable detail on demand. For this we currently add a second window to the application that contains sliders to identify how volatile, complex, or controversial each factor is.

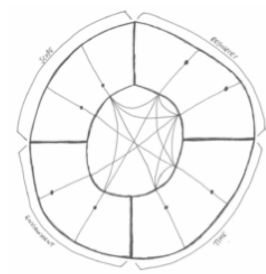

Figure 2: Conceptual sketch of an alternative confluence diagram

Further we developed an alternative approach of the confluence diagram (Fig. 2) in which we visualize the dependencies in a separate area in the inner circle of the diagram. We will implement this alternative confluence diagram in future experiments.

7. CONCLUSION AND OUTLOOK

First trial runs with project managers and business executives have shown that this application serves a clear and pressing need for many meeting contexts. In addition, it seems intuitively understandable to the targeted user groups. The repeated use of the tool will provide significant knowledge management benefits to an organization, as it will be able to analyze the visualized knowledge systematically over time. Still much more user testing and application tests in different contexts are needed to demonstrate the value and versatility of this novel approach to configure project constellations by combining different sectorial expertise.

ACKNOWLEDGMENTS

The authors thank armasuisse, Kuoni and GfK SE for financial support of this study in the context of the Dynagrams project.

REFERENCES

[1] Eppler, M., Sukowski, O. Managing Team Knowledge. European Management Journal, vol. 17, no.3, pp. 334-341, 2000.

[2] Yu-Chun Chang, Chi-Jui Chang, Kuan-Ta Chen, and Ching-Laung Lei. Radar Chart: Scanning for Satifactory QoE in QoS Dimensions, IEEE Netw., vol. 26, no. 4, pp.25 -31, 2012.

[3] M. J. Saary, Radar plots: a useful way for presenting multivariate health care data, Journal of Clinical Epidemiology, 60, pp.311-317, 2008.

[4] Few, S. Now You See It: Simple Visualization Techniques for Quantitative Analysis. Berkeley, CA: Analytics Press, 2009.

[5] Heer, J., Shneiderman, B. Interactive dynamics for visual analysis, acm queue, vol. 55, no. 4, pp. 45-54, 2012.

[6] Hunt, P. React: Rethinking Best Practices. JS Conf EU presentation. http://2013.jsconf.eu. http://de.slideshare.net/floydophone/react-preso-v2 (15.9.2013).

[7] Nolen, D. The Functional Final Frontier. Clojure/West presentation. http://youtu.be/DMtwq3QtddY. (24.3.2014).

[8] Shimojima, A. The Graphic-Linguistic Distinction. Artificial Intelligence Review, 15, pp. 5-27, 2001.

[9] Norman, D. Affordance, conventions, and design. Interactions, May-June, pp. 38-44, 1999.