Concept Mapping: Connecting Educators
Proc. of the Third Int. Conference on Concept Mapping
A. J. Cañas, P. Reiska, M. Åhlberg & J. D. Novak, Eds.
Tallinn, Estonia & Helsinki, Finland 2008

USING CONCEPT MAPS TO ENHANCE SYSTEM VIEW NAVIGATION

Thomas Hubbard, MITRE Corporation[1]*, USA*
Judith A. Stafford, Department of Computer Science, Tufts University, USA
thubbard@mitre.org, jas @cs.tufts.edu

Abstract. If one is to understand how a system meets its quality goals they must understand the elements and the relationships among the elements of the system. These relationships span multiple views of the system. View-based documentation of software architectures improves the ability of stakeholders to understand the system and in particular how the system meets its quality goals [Clements]. However, in order to understand the relationships among elements of different types in a system one must provide a mapping across views. Concept maps are a tool for visualizing relationships between different concepts. In this paper we describe the application of concept maps as a solution to the problem of mapping across different views of software, which in the past, have limited the stakeholder's ability to understand the complex relationships that exist in software.

1 Introduction

If one is to understand how a system meets its quality goals they must understand the relationships among the elements of the system. The elements and relationships among them are communicated using different views of the system. Fundamentally these views all represent conceptual models of the same system and collectively are called the architecture of the system. However, due to the number of different stakeholders and the amount of information to convey several conceptual models of the system must co-exist.

Often times understanding of the system being developed is maintained as part of a program team with individual members becoming experts in parts of the system. According to Smolander (2002) this behavior hints at organizational problems and management urges that this architectural knowledge be spread throughout the organization. To mitigate this risk several processes have been created to provide visibility into the system development process. However, documenting systems is not enough because there are issues with documents themselves. Documents tend to become obsolete over time and ripe with inconsistency (Cederling 2000). The solution to identifying element interdependencies and enhancing stakeholder communication has certain requirements. Due to the complexity of software systems the shared understanding must convey large amounts of information efficiently and effectively in a simple manner. Due to the number of problem domains that software engineering is applied to, any solution must be very expressive and must be able to support many problem domains as well. In order to support the broadest set of stakeholders possible the solution selected must be easy to learn with little or no formal training. The solution should support a collaborative environment so that ad-hoc interpretations can be avoided by performing the work in a group environment. Finally, the shared understanding should not be limited to one phase of the software development life cycle but be incorporated as a fundamental part of the process.

This paper describes the use of Concept Maps (Novak 1998) as the basis for a solution to creating such a collaborative environment. Concept maps will provide the basis for a common view to support a shared understanding of the system. In addition, since the concept map provides a shared view of the system the concept map will act as a bridge to map between views of a system thereby allowing different views of the system to be effectively cross referenced.

2 View-Based Approach

View-based approaches to documentation are used to describe software architectures by giving the various stakeholders different views of the system. View-based approaches are therefore effective tools for allowing specific stakeholders to focus on parts of the system for which they have concerns (Clements 2003, IEEE1471). However, view based approaches suffer from a lack of tools and techniques to effectively map between different views of the system. During meetings where different stakeholder groups come together the views used by the stakeholder groups will be translated into a generic form that can be communicated. For example, the system architect will map the UML view of the system into something the customer will understand. This translated view may or may not be technically accurate but is used for illustrative purposes. The problem with these view

[1] The author's affiliation with The MITRE Corporation is provided for identification purposes only, and is not intended to convey or imply MITRE's concurrence with, or support for, the positions, opinions or viewpoints expressed by the author.

translations are that they are 1) inaccurate due to the subjective nature of the translation 2) inconsistent due to ad-hoc translations 3) time consuming to produce 4) inaccurate when translated from a more robust representation and 5) are out of date since work must be done to keep the 'stakeholder' view in line with the working model. All of these problems result in miscommunications, wasted time and effort.

]

Documenting architectures such that different stakeholders can find the information they need and to understand the information once it is found is a difficult task. For example, Figure 1a and Figure 1c illustrates two UML-based views of the Online Collaborative Lesson Planner system that was developed as part of a pilot study in 2007. The lesson planner is used as part of a course for K-8 teachers was sponsored by the Technical Education Research Center (TERC). The Collaborative Lesson Planner is used in the training of student-teachers to become teachers.

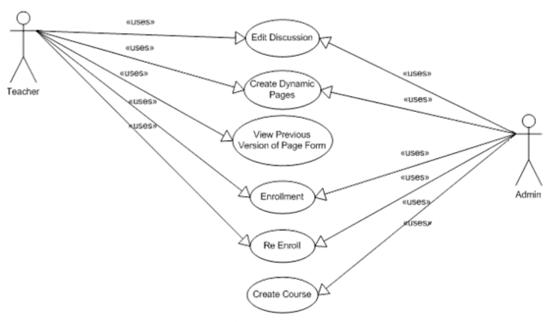
3 Concept Maps

A concept map is a graphical representation of the concepts of a body of knowledge along with their interrelationships in the form of a directed graph (Novak 1981). The map itself consists of a focus question to describe the scope of the map the concepts are represented as nodes and the relationship between the concepts are represented as directed arcs with connecting words. Figure 1b shows a concept map that illustrates the body of knowledge related to the Collaborative Lesson Planner described above. The concept map was created using the process as described by Novak (1998) using input from the Collaborative Lesson Planner Development team in addition to concept maps previously created by the team.

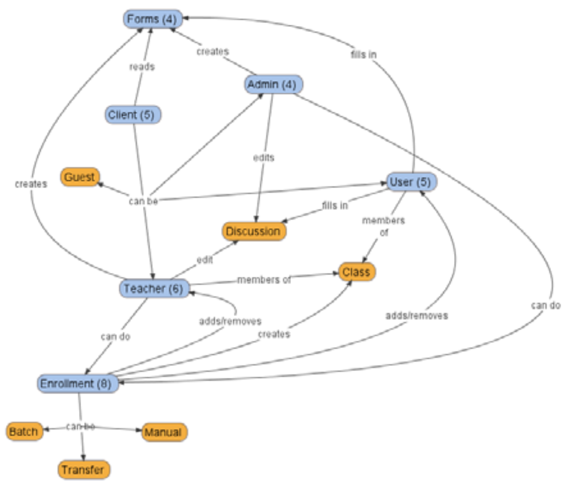
The concept map in Figure 1b differs from the definition offered by (Novak 1998) in the removal of the focus question and the addition of node coloring and degree labeling. The focus question was removed for visual clarity. If one wanted a focus question it could simply be "What does the Lesson Planner Consist Of?" or something similar. The coloring of the Enrollment, Teacher, User, Client, Admin and Forms concepts is used to highlight them as key concepts in the system. Those concepts were selected as key concepts based on the number of relationships that they participate in. This brings us to the degree labeling, the number in parenthesis for each of the blue-colored concepts. The number represents the degree of the node, in other words, the total number of arcs entering and leaving the node. Based on the number of relationships, the concepts Enrollment, Teacher, User, Client, Admin and Forms are ranked in the order of importance. The relevance of the ranking is that during mapping, any part of the implementation that maps to key concepts should be examined for scalability and performance issues since those parts of the implementation will become key parts of the implementation as well. Note, the degree labeling and node coloring are used as a visual aid and can be omitted without changing the idea of the key concepts or rankings.

4 Mapping Activity

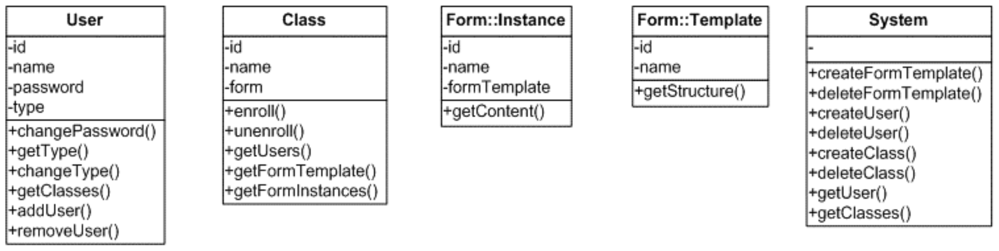
The study selected a use case to class diagram mapping because often times the association of classes in a system involved in the implementation of a use case are not obvious. Mapping between the Collaborative Lesson Planner's use case diagram, concept map and class diagram proved an easy exercise when using a concept map as a bridge. The mapping of the Manual Enrollment use case mapped to the class diagram using the concept map of the system as a bridge is illustrated in Figure 1d. The technique used was a list explicitly describing the relationships between the conceptual models. The list for the Collaborative Lesson Planner's mapping is presented in Figure 1.



Use Case Diagram (a)



Concept Map (b)



Class Diagram (c)

Mapping (d) Use Case	Concept Map	Class Diagram
Manual Enrollment is a specialization of the Enrollment use case.	The proposition "Enrollment can be Manual".	The activity of Manual Enrollment is not identifiable explicitly in the class diagram.
	The proposition "Enrollment creates Classes".	The System class is used to create Class and User objects. User objects represent Users and Teachers.
	The proposition "Enrollment adds/removes Teachers."	Unclear
	The proposition "Enrollment adds/removes users."	Enrollment is performed by calling the enroll method on Class object.

Figure 1 Mapping the Collaborative Lesson Planner's Manual Enrollment Use Case

This mapping technique proved very useful for identifying relationships between the conceptual models of the system and identifying design issues. Evidence for its utility can be seen in the following examples from our experience with the Collaborative Lesson Planner.

First, the concept of User is overloaded. User appears in the concept map as a subordinate to Client but in the class diagrams the class User takes on a more significant role as the base class for all user types. The concept map clearly identified this disconnect from the conceptual model of the system which must be addressed in order to present a consistent conceptual model of the system.

Second, the Enrollment concept has several relationships in the concept map yet enrollment does not appear explicitly in the class diagram of the system at all. Enrollment is hidden in the implementation of several classes: System, User and Class. With such a critical piece of the system hidden it will become difficult for maintenance programmers to fix problems. This will have a direct impact on the maintainability and therefore cost of the system.

Finally, during the pilot study, there was one year lapse between when the first pilot study team ended development and a new study team began development on the same system. Using the concept maps developed by the initial development team, the final development team was able to 1) recreate designs decisions made a year prior 2) communicate the system to new customer stakeholders and 3) continue development with little or no input from the original development team. The pilot team attributed these capabilities to the initial pilot team's use of concept maps. Clearly, these examples show strong support for the use of concept maps as providing a shared conceptual model of the system that can be leveraged in many ways.

5 Conclusions and Future Work

A key goal of this work was to enable a shared understanding between various stakeholders using concept maps for software engineering. Evidence that this goal was obtained was seen in the briefings the pilot study team gave to different stakeholders. The pilot study team was able to bring several stakeholder groups, outside development groups in this case, up to speed on their problem and solution using concept maps as the key discussion driver.

As mentioned previously some of the requirements of any proposed solution are the abilities to present large volumes of information in a compact format and to support many problem domains. During the pilot study, the pilot study teams worked in different problem domains and could represent their architectures in one concept map. Based on that work these two requirements were met. Constructing the artifact in a collaborative environment has been shown to make better designs Additionally, concept maps are effective communication tools as seen in all concept map based pilot study briefings and as such they have been shown to help communicate between different user groups.

Mapping between views proved to be the most difficult aspect of this work. Specifically, the issue of visualizing between conceptual models was the most difficult task. Clearly more work visualizing the actual mapping with more complex systems is needed.

References

Bass, L., Clements, P., Kazman, R., *Software Architecture in Practice, Second Edition.* Addison-Wesley, 2003.

Booch, G., Rumbaugh, J., Jacobson, I., *The Unified Modeling Language User Guide.* Addison-Wesley, 1999.

Cederling, U.; Ekinge, R.; Lennartsson, B.; Taxen, L.; Wedlund, T.. *A project management model based on shared understanding System Sciences.* Proceedings of the 33rd Annual Hawaii International Conference on, Vol., Iss., 4-7, 2000.

Clements, P., Bachmann, F., Bass, L., Garlan, D., Ivers, J., Little, R., Nord, R., Stafford, J.. *Documenting Software Architectures; Views and Beyond.* Addison Wesley, 2003.

Hubbard, T., "Concept Map Based Software Engineering", Tufts University TR-2007-6, 2007.

Novak, J., Applying learning psychology and philosophy to biology teaching. The American Biology Teacher, 43(1). 1981.

Novak, J., Learning, Creating, and Using Knowledge: Concept Maps™ as Facilitative Tools in Schools and Corporations. LEA Incorporated. 1998.

Smolander, K., Pivrinta, T., *Describing and Communicating Software Architecture in Practice: Observations on Stakeholders and Rationale.* Proceedings of CAiSE'02 - The Fourteenth International Conference on Advanced Information Systems Engineering, Toronto, Canada, May 27 - 31, 2002.