Knowledge Modeling in the Health Care Domain to Support Software Development & Maintenance

Thomas Reichherzer, John Coffey, Bilal Gonen and Irad Gillett
Dept. of Computer Science, University of West Florida, 11000 University Pkw., Pensacola, Florida, 32503 U.S.A.

Keywords: Knowledge Modeling, Concept Mapping, Software Engineering, Health Information Systems.

Abstract: This article contains a description of a knowledge elicitation effort and representation pertaining to the modeling of conceptual knowledge in the health care field. The project has the goal of building a conceptual model of data in the Military Health System Data Repository, a large DoD/VA aggregation of databases that can be used in the implementation of software. The goal is to create a just-in-time conceptual model of the data to facilitate software development and foster software developer understanding of the domain.

1 INTRODUCTION

Healthcare information systems play a critical role in providing information about patients, insurers, and service providers. Modern systems have become huge, complex, and difficult-to-understand with no real consistency in the use or meanings of the vocabularies that describe services or service providers. Better integration among health care systems is critical to the goals of cost containment and improved health care delivery. With vast amount of health care data becoming accessible it is important that the transmitted data are well understood by end users and machines in order to support decision making and automatic processing respectively.

Concept mapping (Novak & Gowin, 1984) is a proven technology that helps people express and visualize their knowledge (Briggs et al., 2004; Coffey & Eskridge, 2008; Coffey et al., 2006). Concept maps are graphical representations of propositional knowledge based upon concepts and phrases that make explicit the relationships between concept pairs. This projects aims to develop a knowledge model using concept mapping (Novak & Gowin, 1984; Novak & Musonda, 1991; Turns et al., 2000) as a means to capture domain knowledge pertaining to the concept of "health care provider" as a proof-of-concept to show that:

(1) a concept map-based knowledge model can be an effective tool to develop a domain vocabulary that describes data streams and services generated by health care systems,

(2) an improved understanding on data models can assist software developers to build and maintain integrated systems that can deliver desired services to end users.

The remainder of this paper contains a brief review of literature pertaining to concept mapping and knowledge modeling, and a description of the current work. Included in the description of the current work is motivation for the work, the activities and results that have been achieved to date, and a scheme to assess the utility of the work to the sponsors of the project. The paper closes with a description of anticipated future work.

2 CONCEPT MAPPING & KNOWLEDGE MODELING

Concept maps represent knowledge as a two-dimensional graph capturing concepts and their relationships on a specific topic. In this graph, concepts are visualized as labeled nodes and relationships as labeled links. A concept can be a word or phrase that describes a perceived regularity in events or objects, or records of events or objects (Novak & Gowin, 1984). In contrast, a labelled link includes any word or phrase so that the two concepts connected by a link give rise to a meaningful statement about the concepts. Such a statement is also known as a proposition or a semantic unit that captures human knowledge on a specific topic. Fully developed maps include a cohesive set of concepts

Reichherzer T., Coffey J., Gonen B. and Gillett I..
Knowledge Modeling in the Health Care Domain to Support Software Development & Maintenance.
DOI: 10.5220/0005239004700476
In *Proceedings of the 3rd International Conference on Model-Driven Engineering and Software Development* (MODELSWARD-2015), pages 470-476
ISBN: 978-989-758-083-3

and their relationships to address a specific question that focuses the discussion in the map. Furthermore, a single concept at the top of the map known as the root concept serves as a starting point to explore the captured knowledge of the map.

Concept maps elucidate an individual's conceptualization on a domain, which can be useful to track the individual's learning progress. Novak and Gowin (1984) originally conceived of Concept Maps as a means to externalize conceptual knowledge of science students in service of uncovering what they knew, as well as misapprehensions they held. Subsequently, several groups have embraced Concept Mapping for knowledge elicitation (KE). McNeese et al. (1993; 1995) used Concept Maps to externalize expert knowledge over a variety of knowledge domains. These included the expertise of pilots regarding their decision-making strategies and to create conceptual designs of cockpits, and that of managers, for internal information management systems and procedures. Novak (1998) described knowledge elicitation with Concept Maps as a means of idea generation in groups of people. Beyond the use of concept maps as a learning and teaching tool, successful applications of concept mapping include institutional memory preservation (Coffey & Eskridge, 2008; Coffey et al., 2006), sharing of domain expertise for public outreach (Briggs et al. 2004), visualization of artifacts in Service-Oriented Architecture (SOA) composite applications (Coffey et al., 2012), and representation of software security assurance cases (Snider et al., 2014).

Electronic concept mapping offers the prospect of connecting concept maps via navigational links and annotating concepts with multimedia resources such as video clips, images, and links to Web sites to provide additional information and examples on the concepts in a map. A set of closely-related and interconnected concept maps and their multi-media resources are organized into a knowledge model (Ford et al., 1991; Ford et al., 1993; Ford et al., 1995), a hypermedia representation of domain knowledge. CmapTools (Cañas et al., 2004) is a publicly available software tool for creating knowledge models and publishing them on the Web for knowledge sharing purposes. It offers special drawing and labelling operations for constructing maps as well as a suite of tools for annotating nodes in a map with multimedia resources or links to other concept maps in the same knowledge model. CmapTools is extensively used for research involving concept maps and, generally, as a learning and teaching tool by teachers and students alike (Walker and King, 2002).

Snider et al (2014) describe a study in which concept maps were elicited as a means of efficiently developing security assurance cases. They describe a two-phase process based upon concept map-based knowledge elicitation and modeling. In the first phase, the security expert performs the preliminary work with a KE to identify the critical security concerns, creating a "skeleton" concept map containing nodes for each critical concern. In the second phase, the security expert and KE interview the software developer who demonstrates how he or she has addressed those security concerns. In the process, the skeleton concept map from the first phase is elaborated to document program files, other documentation, and other application responsibilities (such as the server software being configured to produce logs) that address the concerns. The concept map of the software assurance case is augmented with links to the actual files that address the security concerns.

3 THE CURRENT PROJECT

Creating knowledge models typically involves a domain expert and knowledge engineers interviewing the expert and eliciting his/her knowledge and representing it as concept maps (a concise, unambiguous format). The goal of the current project is to conduct a case study that answers questions regarding how knowledge elicitation of domain knowledge via concept mapping might be useful to assist software developers in creating data definitions for programs in an efficient manner. For the project, a new knowledge model was built to help understand the semantics of a complex data model of an actual Department of Defense (DOD) medical information system. The model was then used to develop data type definitions in the form of Enterprise Java Beans. Software developers and domain expert evaluated the process and artifacts that were produced for this case study for usability and completeness.

Motivation for the project comes from the fact that bottlenecks in communication between the domain expert and software developers threaten the timely development of software. The communication problem comes from the differences in expertise between the software developer and domain expert and the inherent complexity and ambiguity of the medical domain in general. Medical terms that label data in healthcare systems lack standardization, requiring interpretation for proper use. The domain expert has deep knowledge about the data and their origin and application in the medical field, while the software engineer understands how to organize data into logical units for efficient access and processing. There is a tremendous amount of information that has to be exchanged to capture and understand data used in healthcare information systems. The lack of knowledge of the software engineer about the domain makes the interpretation of data models very difficult. A concept map-based knowledge model can be an effective tool to develop a domain vocabulary that describes data stored in complex data models. We attempt to show that concept maps can be an effective vehicle for communication between the domain expert and software engineer, giving the engineers relevant information to interpret the data from the data models, alleviating the knowledge gap between software engineer and domain expert. In addition, the possible reuse of built maps may help in breaking down bottlenecks in communication.

4 RESEARCH METHODS

Researchers collaborated with a domain expert who understood the large system of databases named the Military Health System Data Repository (MDR). The MDR is a "centralized data repository that captures, archives, validates, integrates and distributes Defense Health Agency (DHA) corporate health care data worldwide." (Health.Mil, n.d.) The MDR is comprised of more than 95 database tables, each with 50 to 300 attributes, keeping healthcare records to support 9.6 million military and related beneficiaries world-wide.

The current case study involves information retrieval pertaining to providers. The provider topic is of particular interest to this study because of the ambiguities in the representation of provider information. Two simple but realistic scenarios were created around the general problem of matching patients to providers (e.g. finding a family practitioner or finding a surgeon and facility). From this initial effort, two use cases for software development were created:

Use Case 1: *Determine the simple provider workload given the patient history.*

Background: The user gives the system a provider name and the system returns the workload of that provider. The system answers the question: What work load does a given provider have with his current enrollment panels?

Use Case 2: *Given the workloads of the members of a group of providers, to which provider should a patient be assigned?*

Background: The user gives the system patient name and pertinent other information (geographic location, healthcare needs, personal preferences, etc.) and the system recommends a physician in a way that balances physician workload. Patients might need to be assigned to a new provider because they have not yet seen a provider in the area due to relocation, the previous provider leaving the network, etc.

One-sentence summaries of the use cases were used as focus questions (a question intended to keep the concept map creation effort focused on one particular topic so as to avoid inclusion of extraneous information). A concept map-based knowledge elicitation effort aimed at building conceptual models of the data needed in the use cases was completed. The goal of this activity was to identify the actual data from the MDR needed to implement the use case. The output of this phase was annotated concept maps describing the data needed for the implementation of the use case. The annotations were links into the MDR database schema identifying relevant data for the use cases.

After the knowledge model was developed, the software developers set about creating Java beans to implement the use cases. An initial goal was to link the Java beans to the knowledge models and the use cases. This was to be achieved by including URLs and a list of relevant concepts at the top of each bean referring back to the Web-exported concept map. The software developers presented questions about the information in the knowledge model to the domain expert and the KE for clarification. A transcript was used to capture ongoing conversations among these participants.

5 DATA ANALYSIS

A goal of the initial modeling attempt was to determine what information was needed in the concept maps in order to: (1) identify and link to data elements in the MDR that were needed to develop the software that implemented the use cases, (2) provide conceptual information to help developers understand the application domain, (3) identify base elements that might be in a skeleton concept map for such data modeling activity.

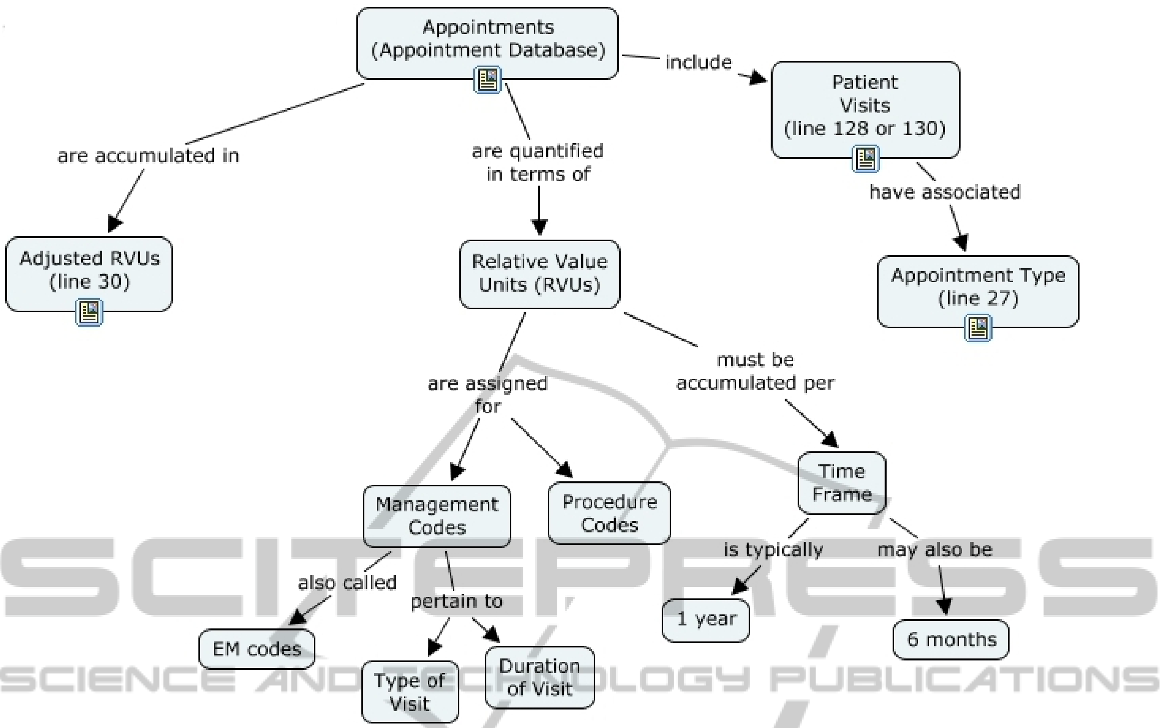

Figure 1: A concept map with domain knowledge and links into the MDA.

The initial knowledge modeling effort produced two concept maps for Use Case 1 and one concept map for Use Case 2 to answer the focus question associated with each use case. The concept map developed for Use Case 2 links back to one of the concepts maps produced for Use Case 1 taking advantage of prior elicited knowledge. Concept maps for each use case were built over two 1-hour modeling sessions that produced progressively more refined maps. The resulting concept maps include for Use Case 1 on average a total of 16 concepts and 17 connections and for Use Case 2 a total of 15 concepts and 16 connections. The initial concept maps were further revised in a follow-up knowledge modeling session after deficiencies in the elicited knowledge were identified during the creation of Java Beans by the software engineer. The revised maps increased in size by an average of 10 concepts and 8 connections for Use Case 1 and 4 concepts and 7 connections for Use Case 2. Figure 1 contains a portion of a concept map from the knowledge model covering information about Appointments. The specific map is part of a larger map created for Use Case 1.

In the course of the knowledge modeling effort several conventions were developed. The first was to indicate when nodes in the concept map pertain to databases. Such a node can be seen at the top of Figure 1, "Appointments (Appointments Database)." The icon at the bottom of that node links to a database schema that describes all elements of that table, the element's data type and format, and some explanatory information.

When specific elements within the database are identified as necessary to the data definitions in the use case, their line numbers are indicated in the concept map. For instance, "Adjusted RVUs" is an important field pertaining to the amount of credit the physician gets for handling an appointment. The "Adjusted RVUs" element is found on line 30 within the Appointments database. The remainder of the concept map contains conceptual knowledge regarding RVUs, specifically, that they are assigned on the basis of management and procedure codes that describe the type and duration of a visit.

As previously mentioned, questions asked by the software developers and the domain expert's answers were captured. Analysis reveals that two general categories of questions were asked and answered:

- low-level technical issues
- higher level conceptual issues.

An example of a low-level question is "What kind of data value does Num(5,3) describe?" An example of a high-level question was "What is the difference between Adjusted RVUs and RVUs in the concept map?" A total of 11 questions and answers were exchanged for the two use cases.

The software developer produced a total of three classes to address the two use cases: `Provider`, `Appointment`, and `Patient`. The `Provider` class had 11 attributes, the `Appointment` class had 4 attributes, and the `Patient` class had 11. The domain expert (who also has knowledge of software development) was asked to judge the quality of the data definitions according to the following criteria:

1. All necessary data is defined to address the use case *or*
2. Some data is missing *and*:
   A. the elements were missing from the conceptual model.
   B. the elements were in the conceptual model but not made explicit enough for the software developer to notice them.
   C. the elements were included in the conceptual model, made explicit, but simply missed by the software developer.

The domain expert judged the data definitions to be sufficient for Use Case 1. However, for Use Case 2, the software developer had created two attributes that were not part of the conceptual model, due to a somewhat idiosyncratic approach to the implementation of containers for the `Patient` and `Provider` classes. However, all needed (and specified) attributes were present both in the conceptual model and in the data definitions produced by the software developer.

6 RELATED WORK

The work described in this paper proposes a new process to support software development and maintenance. It applies concept mapping to capture semantic information of selected healthcare concepts in a just-in-time knowledge modelling effort to address questions related to new use cases. The captured domain knowledge provides context for interpreting information in large-scale, complex healthcare data models. This work differs from a number of other approaches discussed in the literature that use conceptual modeling techniques for system development. Widely used conceptual modeling techniques are the Extended Entity Relationship (EER), the Unified Modeling Language (UML), and variations of it (Combi and Oliboni, 2006). They produce visual data models directly from the narrative of a business use case as abstractions of real-world phenomena of interest. In all cases a specific language must be used to translate the mental representation of a problem domain into a physical diagram. However, as several authors have pointed out in their research (Aguirre-Urreta and Marakas, 2008; Castro, Baião & Guizzardi, 2011), experience in building models, domain expertise, and expressiveness of the modeling language can have a significant effect on the modelling outcome and the usefulness of the resulting model for system development. In contrast, concept mapping requires little training experience for building models and interpreting them. It allows a domain experts to participate in and shape the modeling effort eliminating the need for the software developer as the model builder to have knowledge of the modeling domain or perform interpretation of a narrative that could introduce ambiguity or errors into the model.

In more closely related work, concept mapping has been used to efficiently develop security assurance cases in which pre-built maps were used to facilitate and structure ongoing conversations between software developer and security experts and build comprehensive knowledge models on security assurance cases (Snider et al. 2014). The resulting maps provide evidence as a set of statements and annotations that the software product meets the necessary security requirements prior to its release. However, in the current work concept mapping was used to capture relevant semantic information about healthcare data from a domain expert that is needed by the software developer to create new data definitions and algorithms for processing healthcare data and develop new services.

7 CONCLUSIONS

Several conclusions can be drawn from the current work. First and most important, it was possible to develop the data definitions from the models. An evaluation of the data definitions produced by the software developer revealed that all attributes needed to implement the Use Cases were present. This result was achieved with a minimal amount of information (11 questions and answers) exchanged among the domain expert, KEs and software developer outside of the knowledge contained in the conceptual model.

It was not possible to calibrate accurately the time efficiency of the knowledge modeling effort because quite a bit of initial time was consumed in figuring out what elements should be in a baseline map and how to make the data items in the MDR explicit in the concept maps. Now that these elements are known, future work can include an evaluation of efficiency.

Some conclusions can be drawn regarding the effectiveness of the representation. As evidenced by the questions exchanged among the domain expert, KEs and software developer, the initial knowledge models failed to convey some information both of a high-level, conceptual nature and low-level technical nature. No significant effort was made to address the latter problem. However, KE and domain expert conducted a follow-up knowledge modeling session to address the issue of missing conceptual, high-level information in the initial model, which enabled the software engineer to complete the data definitions for the two use cases. While it seems unavoidable that some conceptual knowledge of interest to the developers might be missed, given the limited amount of time devoted to knowledge elicitation in this study, it would likely have occurred in any representation. The representation used here is well known to be more concise and less ambiguous than textual descriptions. Further, on the basis of the exchanges between the domain expert and the developer it can be determined that, while gaps in the model's conceptual knowledge representations existed, the developers at least knew the right questions to ask.

ACKNOWLEDGEMENTS

Partial funding for this project was provided by Blue-Cross Blue-Shield through the Security and Software Engineering Research Center (http://www.serc.net).

REFERENCES

Aguirre-Urreta, & M., Marakas, G. M., 2008. Comparing conceptual modeling techniques: a critical review of the EER vs OO empirical literature. *The DATA BASE for Advances in Information Systems, 39(2),* 9-32.

Briggs, G., Shamma, D., Cañas, A. J., Scargle, J., and Novak, J. D., 2004. Concept maps applied to Mars exploration public outreach. In Cañas, A. J., Novak, J. D., and González, F., editors, Concept Maps: Theory, Methodology, Technology. *Proceedings of the First International Conference on Concept Mapping*, pages 125-133, Pamplona, Spain.

Cañas, A. J., Hill, G., Carff, R., Suri, N., Lott, J., Eskridge, T., Gómez, G., Arroyo, M., & Carvajal, R., 2004. CmapTools: A knowledge modeling and sharing environment. In Cañas, A. J., Novak, J. D., and González, F., editors, Concept Maps: Theory, Methodology, Technology. *Proceedings of the First International Conference on Concept Mapping*, Pamplona, Spain.

Castro, L., Baião, F., & Guizzardi, G., 2011. A semantic-oriented method for conceptual data modeling in ontoUML based on linguistic concepts. *Conceptual Modeling-ER,* Springer-Verlag, pp. 486-494.

Combi, C. & Oliboni, B., 2006. Conceptual modeling of XML data, In *Proceedings of the 2006 ACM Symposium on Applied Computing*, Dijon, France.

Coffey, J. W., Moreman, D. & Dyer, J., 1999. Institutional memory preservation at NASA Glenn Research Center. *Technical Report,* NASA Glenn Research Center, Cleveland, OH.

Coffey, J. W., & Hoffman, R. R., 2003. A knowledge modeling approach to institutional memory preservation. *The Journal of Knowledge Management, 7*(3), 38-49.

Coffey, J.W. & Eskridge, T., 2008. Case Studies of Knowledge Modeling for Knowledge Preservation and Sharing in the U.S. Nuclear Power Industry. *Journal of Information and Knowledge Management*. 7(3). pp 173-185.

Coffey, J.W., Hoffman, R.R., & Cañas, A. J., 2006. Concept Map-based Knowledge Modeling: Perspectives from Information and Knowledge Visualization. *Information Visualization*, 5. pp. 192-201.

Coffey, J., Reichherzer, T., Wilde, & N., Owsnicki-Klewe, B., 2012. Automated Concept-Map Generation from Service-Oriented Architecture Artifacts. *Proceedings of the 5th International Conference on Concept Mapping.* Valetta, Malta, September.

Ford, K. M., Cañas, A. J., Jones, J., Stahl, H., Novak, J., & Adams-Webber, J., 1991. ICONKAT: An integrated constructivist knowledge acquisition tool. *Knowledge Acquisition, 3*, 215-236.

Ford, K. M., Cañas, A. J., & Coffey, J. W., 1993. Participatory Explanation*. Proceedings of FLAIRS'93, the Sixth Florida Artificial Intelligence Research Symposium*, Ft. Lauderdale, FL, pp111-115.

Ford, K. M., & Bradshaw, J.M., 1995, "Beyond the Repertory Grid: New approaches to Constructivist knowledge acquisition tool development", *International Journal of Intelligent Systems, 8,* 287-333.

Health.Mil, The Official website of the Military Health System and Defense Health Agency. Online, Available from: http://www.health.mil/Military-Health-Topics/Technology/Clinical-Support/Military-Health-System-Data-Repository [14 August 2014].

McNeese, M., Zaff, B., Brown, C., Citera, M. & Selvaraj, J., 1993. Understanding the context of multidisciplinary design: Establishing ecological validity in the study of design problem solving, *Proceedings of the 37th Annual Meeting of the Human Factors Society,* Santa Monica, CA.

McNeese, M., Zaff, B. S., Citera, M., Brown, C. E., & Whitaker, R., 1995. AKADAM: Eliciting user knowledge to support participatory ergonomics.

International Journal of Industrial Ergonomics, 15, 345-363.
Novak, J. & Gowin, D., 1984. *Learning How to Learn*. Cambridge. University Press, New York, NY.
Novak, J. D. & Musonda, D., 1991. A twelve-year longitudinal study of science concept learning. *American Educational Research Journal*, 28(1):117–153.
Turns, J., Atman, C. J., & Adams, R., 2000. Concept maps for engineering education: A cognitive motivated tool supporting varied assessment functions. *IEEE Transactions on Education*, 43(2):164–173.
Walker, J. M. T. & King, P. H., 2002. Concept mapping as a form of student assessment and instruction. In *Proceedings of the 2002 ASEEE Annual Conference and Expo*.
Snider, D., Coffey, J., Reichherzer, T., Wilde, N., Terry, C., Vandeville, J., Heinen, A., & Pramanik, S., 2014. Using Concept Maps to Introduce Software Security Assurance Cases. *CrossTalk: The Journal of Defense Software Engineering*, vol. 27, no. 5.