Computational Linguistic Hedges Applied to a Project Critical Path

Joseph M. McQuighan, PMP
jmcquighan@towson.edu

Robert J. Hammell II, PhD
rhammell@towson.edu

Towson University
Towson, MD 21252

Abstract

Recent research projects have implemented Zadeh's computing with words (CWW) paradigm as a viable solution to estimation issues in the planning stages of projects. This paper moves beyond the planning phase to applying those concepts to the monitoring phases, particularly to the scope constraint, with the goal of controlling activities on a project's critical path. The methodology proposed in this study is to use computational intelligence to implement linguistic hedges that would refine the interpretation of the scope status on a critical path. This paper outlines why scope was selected as the constraint to monitor, why fuzzy sets were determined to be a good tool for the scope constraint, and provides an example as to how computational intelligence would be used to ascertain the scope status for an activity when linguistic hedges are available. By applying hedges that consider the relative importance of the critical path a meaningful status for the scope constraint can be constructed. Since errors in scope lead to cost overruns and schedule delays, the early awareness of scope issues should contribute to rapid corrective actions, thus increasing project success.

Keywords: critical path scope, project scope status, scope linguistic hedges, computational intelligence linguistic hedges

1. PROJECT MANAGEMENT

Information technology (IT) projects go through well defined phases starting with the initiating and planning phases and ultimately moving into an execution phase where the bulk of the implementation work is accomplished. One key to success in managing IT projects is the monitoring and controlling process during this performing and executing phase. Projects are typically constrained by factors such as time, cost, and scope (Gido and Clements, 2009). If the project is drifting from those objectives, then it is important for management to recognize the variances and to make adjustments.

Organizations such as the Project Management Institute (PMI) and the International Project Management Association (IPMA) have published recommendations, tools, and techniques to help practitioners manage projects. One example is the PMBOK published by the PMI, which identifies methods that can be used to manage cost, time, and scope on projects (PMI, 2008). The critical path method is recommended as a technique that can assist in time management (PMI, 2008). For cost management the earned value

©2011 EDSIG (Education Special Interest Group of the AITP)

method calculates cost variances. For the scope constraint techniques such as function points can be used in IT projects in the early phases for estimations. However, scope lacks a precise unit of measure such as currency or time that can be used in the monitoring phase of a project. Fleming and Koppelman (2010) proponents of the Earned Value model have stated that "earned value accurately measures project performance, but must assume that scope definition is adequate."

Scope Issues in IT Projects

When 400 organizations were surveyed on the status of their IT projects it was found that close to a quarter of the reported projects were failures (Levinson 2009a) with only 29% reported as successful. This leaves a significant percent of the IT projects in a less than successful status. With such a high rate of failure, a search for the root causes lead to the conclusion that poor requirements and scope management were contributing factors (Levinson, 2009b). Other researchers and authors support this finding, pointing to cases such as the bankruptcy of FoxMeyer Drug in 1996 due to an IT project that had scope problems (Bulkeley, 1996) or a $170 million project failure by McDonalds Restaurants in 2001 due to scope problems (Youngkuk, 2008).

There are a number of reasons why scope is problematic. According to Weill and Broadbent (1998) projects are late sometimes due to new business needs that occur during the project. This event, called scope creep, impacts cost and schedule (PMBOK 2008). Gido and Clements list a variety of causes for scope problems: the project team might change the design, verbal agreements contribute to misunderstandings of scope, or even a response to the occurrence of events that were identified as risks (Gido & Clements, 2009). Schwalbe pointed out that sometimes from the beginning of a project it is well known that the scope is unclear (Schwalbe, 2010). With so many different sources of scope problems, it is the responsibility of the project manager to quickly identify scope issues, so that corrective actions can be taken.

Scope is much more difficult to monitor because of the lack of an objective unit of measurement on IT projects. Schwalbe stated that managing scope is especially difficult on IT projects because of an inability to verify (Schwalbe 2010), that verification being a word based activity. In a study of complex projects it was found that "soft issues" were hard to measure, but one early warning sign on problem projects was a "gut feeling." (Klakegg, Williams, Walker, Andersen, & Magnussen, 2010) They suggested that additional measures beyond cost and time must be found.

Proposed Solution for Monitoring Scope

If scope is defined in words, then it stands to reason that the monitoring of scope should be comprised of a system that can capture the meaning of words. In the 1990's this was not a realistic possibility, given the limitations of available and accessible computing power. Advances in computing power combined with established mathematics and computer science algorithms enable such a solution to be feasible. Zadeh (2002) proposed that computing with words was possible through the implementation of fuzzy set theory. Other works have proposed that implementing Zadeh's computing with words (CWW) paradigm would be a viable solution to monitoring scope on IT projects (McQuighan & Hammell, 2011). Those solutions can measure and report the subjective scope status.

This paper focuses on the monitoring of scope for activities on a project's critical path. Linguistic hedges can be implemented using computational intelligence to further refine the interpretation of the scope on the critical path. With a tool to capture abstract, nonobjective scope status that takes into account the relative importance of the critical path tasks, the monitoring and reporting can be aggregated into an overall project scope status that more accurately reflects the underlying nature of the project in terms of the set of activities on the critical path.

2. CRITICAL PATH DEFINED

The critical path method is widely used for time management. The PMBOK states that the critical path method calculates the theoretical early start and finish dates by performing a forward and backward pass analysis of the network schedule (PMBOK 2008). This means the critical path is the set of activities that determines the duration of the project, and is the longest path through the project. Managers focus their attention on the critical path, which is usually represented as a network diagram (Gido & Clements, 2009), because deliverables on the critical path that occur on time lead to projects that complete on time. An extremely simple example of a critical path is illustrated in figure 1.

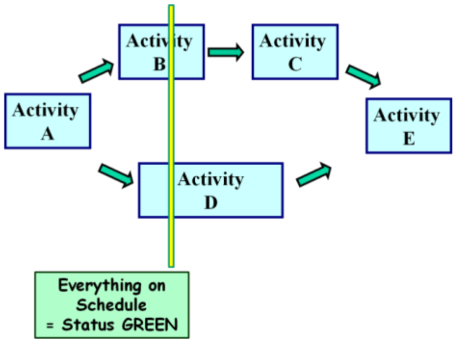

Figure 1. Example of Critical Path

For the purposes of this example, the critical path shall be defined as A-B-C-E, with the current time at the middle of B and near the beginning of activity D. The status of the time constraint would normally be seen as the status of the critical path. Using the Critical Path Method, activity D might have no impact on the project schedule if it runs late (Gido & Clements, 2009).

Critical Path and Project Status

Executives and sponsors tend to focus on problem areas and delays. When issues happen on the critical path, that is an indication that a project is in trouble. It has been shown that project managers tend to underreport project status, especially for critical path activities. Snow and Keil investigated variances between the true status of a software project from the reported status. "The intangible nature of software makes it difficult to obtain accurate estimates of the proportion of work completed, which may promote misperceptions regarding project status" (Snow & Keil, 2001). Snow identified the need for better tools for understanding project status, and the necessity to automate the reporting of status to avoid project manager bias and reporting errors (Snow & Keil, 2001).

Stoplight Reports

For quick problem identification many organizations use a stoplight report (red, yellow, and green) to summarize project status (Dow & Taylor, 2008). Schwalbe gives thumbnail definitions for the color indicators as green = on target, yellow=fair, and red=poor (Schwalbe, 2010). When the project status is not a clear green or definite red, Barnes and Hammell (2008) found that it is difficult for experts to decide that the status of a project is yellow. Furthermore, Snow and Keil (2002) found that IT project status of red is frequently misreported as yellow. What is needed is a system to collect the data and automate the reporting of status using corporate standards. The solution proposed in this paper is an attempt at automating the collection and reporting of scope status. Stoplight reports are common and useful, so the proposed solution feeds into the red-yellow-green system.

3. COMPUTATIONAL INTELLIGENCE

The field of computational intelligence (CI) is a branch of engineering that defines a revolutionary model for evaluating fuzzy, inaccurate inputs. Along with advances in soft computing techniques the automation of the handling of vague and imprecise data is now easily possible. The IEEE Computational Intelligence Society defines CI as a number of core technologies, among them fuzzy systems, neural networks, evolutionary programming, and genetic algorithms (IEEE 2011). The IEEE website states that these technologies build intelligent systems to help with complex problems in which the information and data are vague, approximate, and uncertain.

Fuzzy Sets

From a wide palette of alternatives in the field of computational intelligence, we selected fuzzy logic as the best match to capturing the status of a project's scope. The field of fuzzy set theory originated with Lotfi Zadeh's (1973) proposal of the concept of graded memberships. Graded memberships model the grey areas in the real world that humans easily comprehend (McNeill & Freiberger, 1993). Fuzzy sets are often compared to conventional binary (crisp) set theory. The difference is that with fuzzy sets the boundaries are not clean, they are continuous with membership not limited to the binary 0 or 1. Examples are words that span a range of possibilities such as late or early, or even colors such as green or yellow. McNeill and Freiberger (1993) pointed out that adjectives and adverbs allow for degrees of membership, and that people accommodate and reconcile the vagueness in many situations through the use of hedges.

Fuzzy logic can be contrasted to conventional bivalent logic, which works on exact numbers, intervals, or probabilities. Rather than the hard,

©2011 EDSIG (Education Special Interest Group of the AITP)
www.aitp-edsig.org

crisp nature of binary logic which has been prevalent in the computing industry since the 1960's, fuzzy logic is implemented in "soft computing" techniques, which recognize the graded or grayscale memberships. McNeill and Thro defined the characteristics of fuzziness as: word based, not number based, nonlinear and changeable, and analog (ambiguous), not digital (yes/no) (McNeill & Thro, 1994). All of these are also characteristics of the scope of a project. For this reason, fuzzy sets are a good match for monitoring scope status.

The implication for monitoring scope on IT projects is that fuzzy systems are capable of replicating human decision making. This includes handling vague data, to the point of coping with noisy and/or missing data (Yen, & Langari, 1999). Another benefit of using fuzzy systems is that they allow for the possibility something can be both "true" and "false" at the same time (Zimmerman, 1996). This is an extreme case, but with the measurement of scope being inherently fuzzy, it makes sense to select a branch of mathematics that can cope with contradictory input, such as might be found with scope status.

Computational Intelligence on Cost and Time Constraints

Computational intelligence has been applied to time and cost constraints by researchers. Even though those constraints have objective criteria that are measureable quantities, there can be fuzziness in the interpretation of those numbers. Li, Moselhi, and Alkas created a forecasting method for cost and schedule constraints using Fuzzy Logic to compensate for the variability found on construction projects (Li, Moselhi, & Alkas, 2006). Their methodology used fuzzy logic for project forecasting and status for the otherwise concrete constraints.

Other researchers have applied computational intelligence tools to project management for schedule control. As recently as 2007 Wang and Hao proposed a Fuzzy Linguistic PERT (Program Evaluation & Review Technique) to replace stochastic methods, storing activity durations as fuzzy sets (Wang & Hao, 2007). Their emphasis was on the imprecision in time estimates. This paper proposes a novel approach going beyond the time and cost constraints, and addressing the scope issues so prevalent on projects.

Computational Intelligence for the Scope Constraint

As part of their detailed analysis of complex project on a variety of settings, Klakegg and his team summarized the criteria that they found could be used as measurements that would ensure project success. At the same time they conceded that warning signs of problems are often unclear and imprecise (Klakegg, et al., 2010). They acknowledged that conventional measurements were not a complete solution, and that squishy, soft human intuition sometimes provided better insight to the true status of a project. To handle the vagueness inherent in gut feeling warning signs, we propose using a CI tool to capture scope status in a more realistic, human friendly manner, and in a form that can be implemented with proven technologies.

An issue in managing scope on software projects is that people attempt to precisely measure that which is not truly measureable. A software developer is only putting forth an opinion that they are 90% complete. Practical experience has clearly recognized a software status of exactly 90% as highly unlikely, and is recognized by project managers as an indicator of potential problems (Gerosa, 2003). Zadeh has stated that "it is a common practice to ignore imprecision, treating what is imprecise as if it were precise" (Zadeh 2010). Zadeh suggested that by using fuzzy set theory, a fuzzy data collector could be created to capture subjective inputs. Zadeh illustrated his concepts as: conventional binary logic would use a sharp pencil to draw a clean line. Fuzzy sets would instead use a can of spray paint (Zadeh 2010).

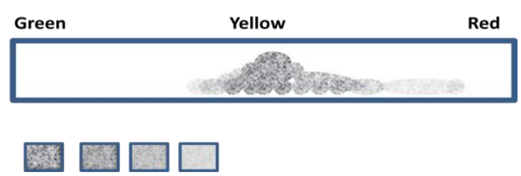

Figure 2. Scope measured using the Z-mouse

Using that analogy Jose Barranquero and Sergio Guadarrama implemented Zadeh's computing with words concepts as computerized spray paint for verbal scales (Barranquero & Guadarrama, 2010). They created a computerized web gadget which they call the Z-mouse to gather fuzzy opinions, or perceptions, from users (Barranquero & Guadarrama, 2010). This study builds upon their prototype by evaluating the fitness of their Z-mouse concepts when applied to project management. Project managers are asked to rate the scope for a WBS activity on a scale that is words, not numbers. The non-numeric scale should be easy to use by experienced project managers. Figure 2 illustrates the Z-mouse web gadget using a non-numeric, linguistic scale. The person responsible for reporting the scope status for an activity would spray paint their impressions or judgments using the web gadget.

These spray paint data points are converted to numeric values, and then evaluated using the strict mathematical rules of fuzzy sets. These numbers are then associated with and stored as part of an activity. The benefit is that the reported status is now a permanent part of a monitoring system. Using computational intelligence algorithms the status can be tracked, evaluated, and summed with the status of other activities (McQuighan & Hammell, 2011)

CI has well defined rules for aggregating multiple opinions using established mathematic rules based on Zadeh's original definitions. Details are described in separate works by Klir, St. Clair, & Yuan (1997) and Zimmermann (1996). Going back to Snow's finding that project managers tend to report optimistic status and downplay the early warning signs (Snow & Keil, 2001) capturing the early warning signs in a computer or centralized database leaves a persistent record in a format that cannot be ignored. The scope status would be a measurement that is analogous to cost and schedule measurements. Since errors in scope lead to errors in cost and schedule, the awareness of scope problems should contribute to early corrective actions, increasing project success.

4. LINGUISTIC HEDGES

In the English language when describing the status of scope, verbal hedges are quite common. Adverbs and adjectives are used to modify or clarify the base meaning of a term. If a project manager asks if a software developer has completed a task, a common reply might be "mostly done." The project manager wants a clean, crisp decision and instead gets the hedge of "mostly." Other hedges include "somewhat", "rather", "nearly", or "almost." The English language contains numerous hedges, and it represents the human ability to distinguish, separate, and attempt to communicate shades of grey and nuances.

In the field of computational intelligence hedges are words that modify fuzzy sets. McNeill and Freiberger explain that hedges operate on fuzzy sets by creating subsets (McNeill & Freiberger, 1993). The word "very" concentrates or shifts a set, making a subset. A cost constraint that is very over budget has moved from the set of "over budget" into the subset of "very over." McNeill categorizes other hedges into groups. *Contrast intensification* hedges separate (very, extremely), *quantifiers* help define (most, several, or few), and *truth values* nudge in a direction (quite true, mostly false.)

Kosko likened hedges to weights that are some degree from 0% to 100% that strengthen or weaken. His fuzzy cognitive map can be used to predict when there is a causal link or connection, with the hedges providing an increase or decrease. (Kosko, 1993). Verbal hedges "a little" or "somewhat" or "more or less" are vague, but computational intelligence tools have defined rules to handle the fuzziness.

In project management an activity might appear as a set of requirements that are "not totally" understood. The hedge *not totally* takes the state called *understand requirements* and modifies the scope status by just a little in the direction of confusion. What was a crisp set of understood requirements is now a subset that is fuzzy. The quantity *not totally* is imprecise, and acts as a qualifier for the base state "understand requirements."

Computational Intelligence tools mimic the English language, and can be exploited as applied computer applications. For the purposes of the example shown below, only three hedges will be used for simplicity. A complete production grade CI system could span a more complex range of possibilities. Applying computational intelligence to scope on a project would mean that a project manager would create a set of hedges, and then use computational intelligence techniques to adjust the reported status for an activity. The fuzzy status gathered with the Z-mouse would be tempered by the appropriate hedge. An example is in order.

Computational Intelligence Hedges

For the purposes of this example, the scope constraint might have hedges that weight the relative importance of the scope for a given an activity as one of three criteria: very important, neutral, or less important. Figure 3 illustrates the fuzzy qualifier "very" as a curve. Cox calls these Sigmoid or Logistic representations, or the easier to remember name: S-curves (Cox, 1999). The role of the S-curve for the hedge "very" is to increase the importance of a subjective evaluation.

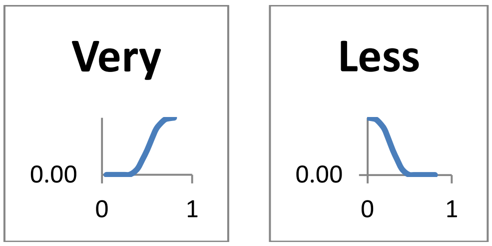

Figure 3. Hedges "very" and "less"

Figure 3 includes a curve for the hedge "less" which is used to diminish the importance of the subjective evaluation. For the hedge "neutral" no modification will be made to the subjective evaluation.

$$S(x: \alpha, \beta, \gamma) = \left\{ \begin{array}{ll} 0 & \text{-->}\ x \le \alpha \\ 2*((x-\alpha)/(\gamma-\alpha))^2 & \text{-->}\ \alpha \le x \le \beta \\ 1 - 2*((x-\gamma)/(\gamma-\alpha))^2 & \text{-->}\ \beta \le x \le \gamma \\ 1 & \text{-->}\ x \ge \gamma \end{array} \right\}$$

The formula to derive the Sigmoid curves is taken from Cox, who defines the inflection point **β**, as the point where the domain value is 50% true. The **β** is selected to put the curves close to the corresponding ends. The **α** and **γ** represent the extremes, with **α** as the zero membership value, and **γ** the complete or 100% membership value.

The selection of **α**, **β**, and **γ** would be determined by an appropriate authority, such as the project manager or a project management office. These computational qualifiers could then be used in mathematical operations to modify scope, a process which will be detailed in the next section.

5. SCOPE ON THE CRITICAL PATH

With definitions of the computational hedges in place, and an established critical path for a project, and a mechanism for gathering the subjective opinions on the status of scope, the next step is to pull all of these pieces together to create a status for the scope constraint. The normal use for a critical path is to manage the time constraint. Activities that exceed their allocated duration cause project problems. Since scope is the source of many project problems, adding information about scope to the critical path activities provides new insights.

The proposed system of gathering fuzzy status with computer tools offers new opportunities for the project manager. It allows the project manager to adjust the interpretation of the scope status for an activity using the CI hedges. A project manager could use the hedge "very important" to intensify the reported status for activities on the critical path, and to diminish the importance of reported status for activities not on the critical path with the "less" hedge. The decision as to how much or little might be policy dictated by a project management office or a project sponsor.

Figure 4 contains an example of a project with five activities. The critical path is through activities A-B-C-E. The current time is in the middle of activities B and the beginning of D. The assumptions are that an appropriate authority has defined and authorized the CI linguistic hedges, and that a scope status has been input with the fuzzy data collector.

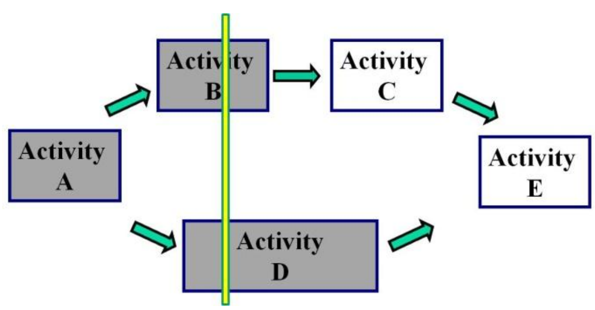

Figure 4. Sample Critical Path

For activity C which has not started, a subjective opinion might report scope status of YELLOW with a leaning toward a scope status of RED, as shown in figure 5.

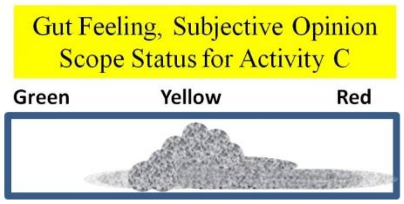

Figure 5. Sample Subjective Opinion for Activity C

For activity C this would feed into the computational intelligence tools. With the hedge "very important" intensifying the reported status for activities on the critical path, the CI tools might result in a scope status of RED for an activity on the critical path, such as shown in figure 6.

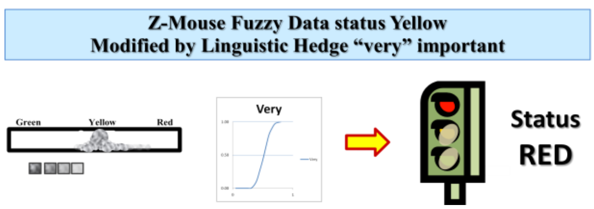

Figure 6. Applying Linguistic Hedges to the Critical Path

The goal is to use the CI linguistic hedges to raise attention to problems with the scope on an activity. This would give the project manager the opportunity to apply corrective action to that element of the activity that is the source of the problem. Using conventional techniques, such as earned value, the problems with the scope would show up after time and money had been expended. The scope status can be captured as a subjective opinion much sooner.

Scope as Predictor

If activity C has not started, using conventional tools, the only critical path information might be that predecessor activities are on time, early, or late for the time constraint. Similarly, a technique such as earned value would give insights to the financial status of an activity, after the funds have been spent. However, for large complex projects it is common to use rolling wave decomposition, which means that the requirements and scope for activities not started have not been defined (PMI, 2008). Yet, it is possible to report in advance that people do not feel comfortable with the scope of a future activity when using a tool such as the fuzzy Z-mouse.

This gut feeling or comfort level is the nebulous quantity that Klakegg stated need to be captured (Klakegg, et al., 2010). With conventional methods, scope status was word based reports that were subject to the interpretation of project managers. As already pointed out (Snow & Keil, 2002) project managers tend to be optimistic, and thus lose the real status that should be of concern. By monitoring scope with computational tools, a new capability can be added to the project assessment. Rather than the retrospective monitoring that is in common use, the system captures the predictions of the humans inputting the data.

In the example above, the software developers might anticipate that activity C has poorly defined scope, or has scope issues. Humans look at future activities and offer their insights as to the status of the scope constraint for an activity that has not started. This gives a project a forecast or predictive status to add to the conventional quantitative status reports on cost and schedule. The CI tools capture scope status predictions from humans and report that information. The CI tools provide the means for collecting and preserving the predictions. Scope as a predictor has traditionally been ignored because it was fuzzy or subjective, and not measureable with tools available in the past.

Other Considerations

This proposed system is only concerned with capturing scope status and reporting that status at the given point in time. Future work might allow for the tracking over time of the scope status. This would be analogous to the way that earned value charts illustrate the financial progress of a project. It might be possible to chart scope status in a similar manner. If the scope status for an activity is slowly slipping from green to yellow to reddish-yellow over time, then this should raise management concerns. At this time, this research is limited to reporting current status, and weighting that status if it is on the critical path.

6. CONCLUSIONS

Through the use of computational linguistic hedges the assessment and monitoring of information technology projects can be enhanced. The tools proposed are an application of the computing with words paradigm (CWW) proposed by Lotfi Zadeh. As Mendel and Wu stated "Zadeh did not mean that computers would actually compute using words... (but that) computers would be activated by words" (Mendel & Wu, 2010). The words would be translated or encoded into fuzzy sets using strict mathematical rules. The fuzzy sets would be processed by a CWW engine, and then decoded into a solution in the form of human understandable words (Mendel & Wu, 2010).

Our next steps are to make a prototype system and use project manager “subject matter experts” to provide input for test scenarios and to validate its performance. Augmenting the traditional critical path method normally used for time management with verbal hedges, and associating it with a fuzzy status for the scope constraint produces an additional tool that can be used to control projects.

7. REFERENCES

Barnes, A., & Hammell, R. (2008). Determining Information Technology Project Status using Recognition-primed Decision Making Enabled Collaborative Agents for Simulating Teamwork (R-CAST). *Proceedings of CONISAR 2008.*

Barranquero, T., & Guadarrama, S. (2010). "Collecting Fuzzy Perceptions from Non-expert Users," *IEEE World Congress on Computational Intelligence July, 18-23, 2010 Barcelona, Spain.*

Bulkeley, W. (1996). Technology (A Special Report): Working Together --- When Things Go Wrong: FoxMeyer Drug took a huge High-tech gamble; It didn't work. Wall Street Journal. Retrieved July 16, 2011 from http://homepage.cs.uri.edu/courses/fall2007/csc305/Schedule/FoxMeyer3.pdf

Cox, E. (1999). The Fuzzy Systems Handbook. AP Professional, New York.

Dow, W., Taylor, B. (2008). Project Management Communications Bible. Wiley Publications, Indianapolis, IN.

Fleming, Q., & Koppelman, J. (2010). Earned value project management. Project Management Institute, Inc., Newtown Square, PA. p.139.

Gerosa, S. (2003). Earned Value Management: How to avoid the 90 percent complete syndrome. Retrieved 29 July 2011.

Gido, J., & Clements, J. (2009). Successful Project Management, Fifth Edition. South-Western Cengage Learning, Mason, Ohio.

IEEE Computational Intelligence Society statement of scope. Retrieved on 30 June 2011 from http://ieee-cis.org/about_cis/scope/.

Klir, G., St. Clair, U., & Yuan, B. (1997). Fuzzy Set Theory. Prentice Hall PTR, Upper Saddle River, NJ.

Klakegg, O., Williams, T., Walker, D., Andersen, B., & Magnussen, O. (2010). Early Warning Signs in Complex Projects, Project Management Institute, Inc. Newtown Square, PA.

Kosko, B. (1993). Fuzzy Thinking. Hyperion, New York, NY.

Levinson, M. (2009). How to Spot a Failing Project. *CIO Magazine.* Retrieved 5 July, 2011 from http://www.cio.com/article/124309.

Levinson, M. (2009). Project Management The 14 Most Common Mistakes IT Departments Make. CIO Magazine. Retrieved 5 July, 2011 from http://www.cio.com/article/438930.

Li, J. Moselhi, O., & Alkas, S. (2006). "Forecasting Project Status by Using Fuzzy Logic." *Journal of construction Engineering and Management, Vol. 132, No. 11, November 2006.*

McNeill, D., & Freiberger, P. (1993). Fuzzy Logic. Simon & Schuster, New York.

McNeill, F. Martin. Thro, E. (1994). Fuzzy Logic. AP Professional, Boston, MA.

McQuighan, J., & Hammell, R. (2011). Published in MAICS Proceeding April 17, 2011. *Midwest Artificial Intelligence and Cognitive Science Conference 2011*.

Mendel, J., & Wu, D. (2010). Perceptual Computing: Aiding People in Making Subjective Judgments. IEEE Press John Wiley & Sons, Hoboken, NJ.

Project Management Institute (2008). A guide to the project management body of knowledge (PMBOK guide) (Fourth ed.). Project Management Institute, Inc., Newtown Square, PA.

Schwalbe, K. (2010). Revised An Introduction to Project Management, Third Edition. Kathy Schwalbe, LLC, Minneapolis, MN.

Snow, A., & Keil, M. (2001). The Challenge of Accurate Software Project Status Reporting: A Two Stage Model Incorporating Status Errors and Reporting Bias. *Proceedings of the 34th Hawaii International Conference on System Sciences.*

Snow, A., & Keil, M. (2002). "A Framework for Assessing the Reliability of Software Project Status Reports*." Engineering Management Journal, June 2002*.

Wang, J., & Hao, J. (2007). "Fuzzy Linguistic PERT." IEEE Transactions on Fuzzy Systems, vol. 15, 2.

Weill, P., & Broadbent, M. (1998). Leveraging the New Infrastructure. Harvard Business School Press, Boston, MA.

Yen, J., & Langari, R. (1999). Fuzzy Logic: Intelligence, Control, and Information. Prentice Hall, Upper Saddle River, NJ.

Youngkuk, Y. (2008). McDonalds Innovate Project failure. *PMINIT November 3, 2008.* Retrieved July 6, 2011 from http://pminit.blogspot.com/2008/11/mcdonalds-innovate-project-failure.html

Zadeh, L.A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics*, 3, 28-44.

Zadeh, L.A. (2002). From computing with numbers to computing with words. *International Journal Applied Mathematics and Computer Science Vol. 12, no. 3*.

Zadeh, L.A. (2010). Precisiation of meaning—toward computation with natural language. *2010 IEEE International Conference on Information Reuse and Integration.*

Zimmermann, H.-J. (1996). *Fuzzy Set Theory and Its Applications, third edition.* Dordrecht, NL: Kluwer Academic Publishers.