

A Project Risk Assessment Model

Predictive Project Risk Assessment and Risk Classification using a Fuzzy Case-Based Reasoning System

©1999 Earl Cox



1289 North Fordham Blvd. Suite A312
Chapel Hill, NC 27517

(919) 678-0477
www.scianta.com

Ergo Cogito Sum!

Therefore, I Think I am.

With further pardons to Rene Descartes (1596-1650)

Project Estimation – The act of forecasting the future by completely ignoring the past.

Project Management – An oxymoron.

From, *A Dictionary of Project Management*

A private publication ©1984 Earl Cox

From early 1974 until 1986 I was founder and president of Interactive Logic, a New York based software company that developed as well as marketed an advanced project management and enterprise modeling system through an international timesharing company. Our project management software, SRMS, was based on a fully relational database model as early as 1976 and included, by the end of the 1970's, backward and forward chaining machine reasoning techniques to allocate resources and design project precedence networks. In the very early 1980's we added fuzzy logic to our system, providing managers and project leaders with the ability to construct fuzzy critical path networks, perform fuzzy project risk and sensitivity analysis, and a use a fuzzy selection and clustering component to SRMS' own query language. Over the past fourteen years I have continued to develop advanced project risk and complexity projects for my clients. This article is based on my experience in actual risk assessment and configuration projects.

Project management remains one of the major hurdles in applied machine intelligence, not to mention allied fields such as operations research , statistics, and mathematical programming. The central issues in project management revolve around two concepts: the accurate estimation of project parameters and management of the project completion process. The first, obviously, has a profound affect on the second. In fact, the inability to come to terms with the vagueness and uncertainties in project parameter estimation continues to plague management's ability to recognize the inherent risk of assuming some projects. Understanding this risk is critical in evaluating both the dangers of committing to a collection of projects, as well as in applying sound project yield metrics – that is, specifying the expected return on investment for high risk projects versus the opportunity costs associated with other less risky projects. Naturally much of these decisions are coupled to the risk aversion or risk tolerance culture of the corporation.

In this article we will explore a system for analyzing and hence predicting the risk of a project based on its structure. The system employs a process known as Case-Based Reasoning (CBR). With a CBR approach we analyze the performance of past projects (the database of cases or exemplars) to predict the performance of a currently planned or proposed project. Other uses of the CBR repository include the semi-automatic generation of a project activity plan, the calibration and refinement of estimates, the re-configuration of a project plan, assignment of or candidate recommendations for the project manager, and the allocation of critical resources,

Project Property Measurements

Evolving a risk assessment for a project involves a wide spectrum of complex parameters. In evaluating a project's risk, many of the important parameters are measures of the change in the values from

the initial project specification This is, in fact, what we are ultimately predicting – the possibility that, given a basic project configuration C_i , that the critical parameters associated with C_i will have an unacceptable slippage. Figure 1 illustrates many (but not all) of the important variables in a project risk assessment application.

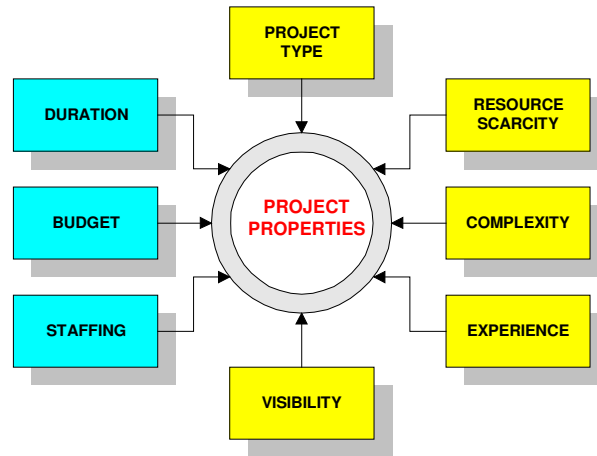


Figure 1. Principle Project Attributes (or Variables)

The parameters on the left (in blue) are the core attributes of a project. They consist of several components: the original estimate, the actual (final) value, and the number of revisions to the parameter. From the original and actual values we can calculate the magnitude of the change (negative for a final amount less than the original and positive for a final amount greater than the original.) The remaining parameters (in yellow) are descriptive attributes that help categorize the project. Many of these are measured on a psychometric or arbitrarily ranked scale, usually in the range [0,10]. Zero means that the attribute has minimal or no properties of the concept, ten means that the attribute is completely within the scope of the concept.

- **Project Type** consists of three attributes. The first, on a scale of [0,10] defines the project's priority relative to all other currently authorized (and, perhaps, budgeted) projects. This second attribute defines the kind of project (traditional IT system, special request, advanced technology, etc.) The kind of project classification is usually decomposed into eight to twelve categories. The third attribute reflects the status of the project and includes such enumeration types as: delivered, cancelled (over budget), cancelled (past delivery date/failed to deliver), cancelled (by end user without prejudice), refused by end user, failed to meet end user requirements, etc.
- **Resource or skill scarcity** is a numeric attribute in the range [0,10] indicating the degree to which the critical resources in the project are drawn from a very limited pool. As an example, a project that requires a Pascal compiler designer and an XML (Extensible Mark-up Language) expert may rank [7] if there are few individuals in the organization with these skills. Scarce resources increase project risk since (a) they are difficult to acquire, (b) they are usually in demand across multiple projects, and (c) they often have an above average attrition rate within an organization.
- **Complexity** is a numeric attribute in the range [0,N] reflecting either the size of the project or the connection density of the project. The original and final complexity values for a project are maintained by the system. Either the size or the connection complexity form can be used, but only one approach must be applied to all projects in the history (case) file. The project size attribute is the easiest to use

and is generally calculated as the sum of the activities per project task divided by the number of tasks. This is reflected in the following expression,

$$C = \frac{\sum_{i=1}^{N_T} A_i^T}{N_T}$$

The size complexity value is thus the average task activity count. The connectionist attribute, on the other hand, measures how the project precedence network tightly or loosely connects the activities. As an example, Figure 2 shows a simple critical path network's precedence relationships (duration times are omitted).

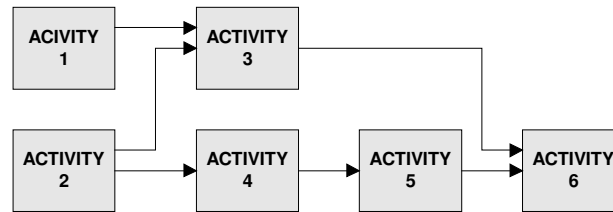


Figure 2. A Simple Project Precedence Network

For every activity in the project, we take the product of the precedence links flowing into and out of it. The complexity is the ratio of the difference between the inflow and outflow products of each activity and the total number of tasks (if either the inflow or outflow is zero, we take the value of this node as one [1]). Expressed mathematically as,

$$C = \frac{\sum_{i=1}^{N_T} \sum_{j=1}^{A_T} \min(1, if \times of)}{N_T}$$

The network shown in Figure 2 has a single task unit with six activities and a connection complexity of [9]. Thus it has a fairly low precedence complexity. However, as the number of dependencies in a project segment grow, the measure of precedence or edge complexity increases rapidly. The connectivity complexity parameter is very useful but is also quite difficult to compute. Generally I use the size complexity measurement unless the bulk of the client projects have complex precedence relationships with little free float (in which case the connection complexity often yields a better measure of the project's intrinsic brittleness.)

- **Experience** is a numeric attribute in the range [0,10] indicating the degree to which the organization has had experience with similar projects in the past. Experience not only creates a categorization partition for the CBR engine, but also provides a valuable analytical checkpoint for other project CBR functions (such as estimation error correction and refinement – if, for example, we are consistently late on projects for which we have a reasonable amount of experience.)
- **Visibility** is a numeric attribute in the range [0,10] indicating the degree to which the project has “political” visibility in the organization. Visibility has its own special stresses on a project and must be

included in all systems that attempt a ranking of a candidate project's ultimate risk. From past experience we can almost write a single rule about this attribute:

If Visibility is High then ProjectRisk is Increased

In any case, visibility, like experience, provides a segmentation plane (or categorization partition) for the CBR engine. This is an important attribute in the risk assessment system.

A Project Risk Assessment System

The risk assessment application consists of two core processes: a feature extraction methodology which performs a compact rule induction from the case base and a fuzzy clustering technique which organizes the case base into similar (but not necessarily unique) categories. Clustering is performed each time a new set of cases is added to the case database. Figure 3 provides a schematic representation of how the risk assessment CBR application works.

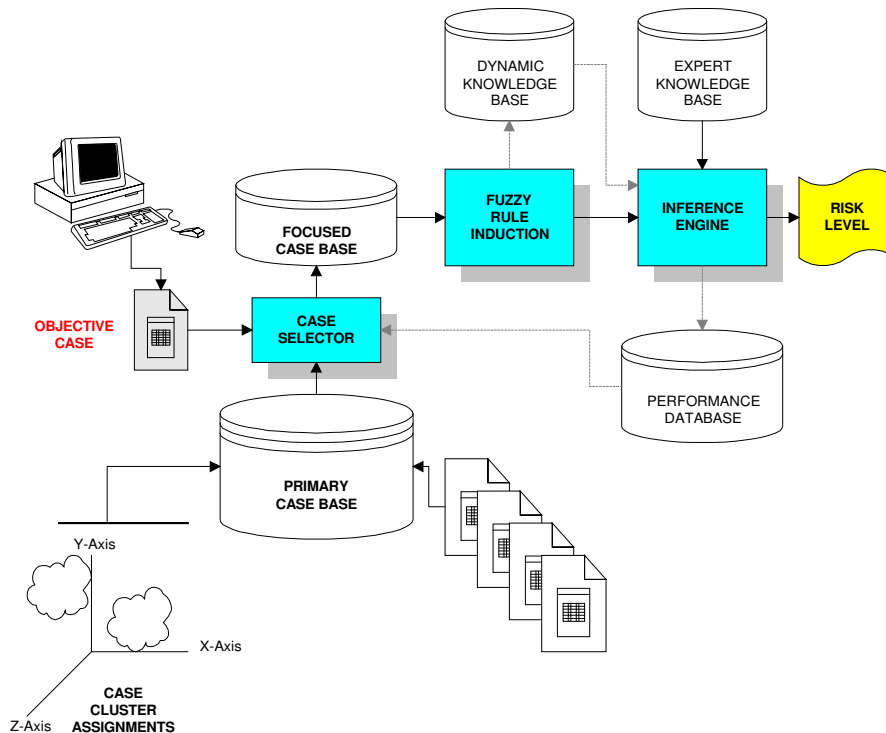


Figure 3. The Basic Risk Assessment CBR Flow

Thus, the attributes of the objective case (the project whose risk we want to assess) are used to select the cases in the correct slice partitions (as an example, project kind, experience, and visibility). These cases are passed through a rule induction process which extracts their features or patterns into a set of fuzzy **if-then** rules (note also that the induced rules can also be combined with expert knowledge to provide a high degree of specificity where subject matter expert (SME) knowledge is important). The rules are run subsequently executed by a fuzzy inference engine to generate the predicted risk assessment ranking (on a psychometric scale in the range [0,10]). A performance database records the objective function, the case

numbers that were selected to generate a risk, as well as the actual risk level itself. It is used to determine how well the application predicts project risk by comparing predicted risk against actual project performance.

Parameter Similarity Measurements

At the heart of a case-based reasoning system is the selection of cases that are similar to the current objective case. How do we measure such similarity? Conventional CBR approaches used a combination of interval arithmetic and space congruity (such as K-nearest neighbor clustering). The fuzzy CBR approach also uses clustering to partition exemplars (members of the case base) into sets with various degrees of membership. But the selection mechanism also employs a form of fuzzy similarity metrics to determine the *degree* to which one case's parameters are similar to another. All numeric case parameters are treated as bell-shaped fuzzy numbers. The width of the bell-curve determines the latitude we have in considering a data point to be representative of the parameter's central value. As Figure 4 illustrates there are two methods of similarity measurement.

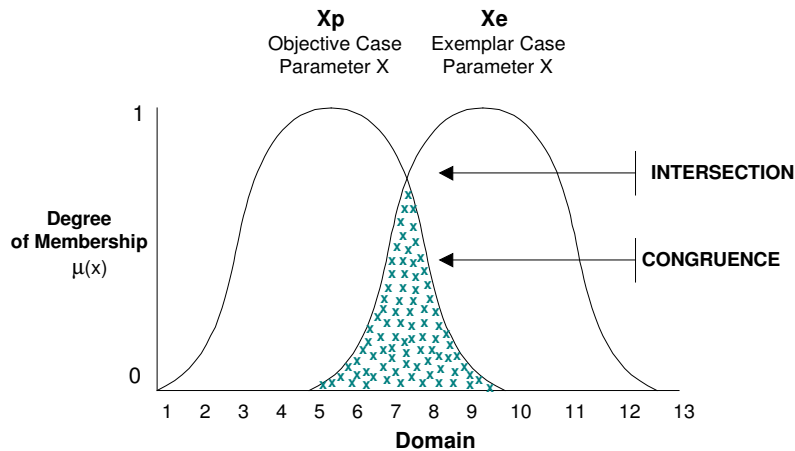


Figure 4. Methods of Similarity measurement

The *intersection* method is quick and effective for isomorphic fuzzy sets (those with the same shape). It measures the degree to which two isomorphic fuzzy sets overlap. The point of intersection is the degree of similarity. When the fuzzy sets do not have the same shape (one is wider or thinner than the other) the *congruence* method is used. This technique takes the ratio of the area shared by the two fuzzy sets and the sum of the areas occupied by both fuzzy sets. By using fuzzy similarity methods, cases can be selected based on the degree to which they match the objective case. This degree is in the fuzzy interval [0,1], and by the extension principle, when a degree of zero is selected, all cases are chosen, and when a degree of one is specified, only cases that are nearly an exact match with the objective are chosen. By working in the interval space $[>0,<1]$ cases that approximately match the objective case are found. These cases generally provide a much richer and more robust solution set. The Fuzzy CBR engine, as it searches for a solution, will automatically relax or tighten the minimum similarity degree for the selected cases in each cycle of the search process.



Learning From Our Mistakes

Understanding risk is essential. Corporations and government agencies attempt to reduce risk through the application of proactive techniques such as precedence (PERT and CPM) analysis, cook-book project methodologies, and prototyping. These approaches do little if anything to aid in the definition of the underlying risks associated with a project based on the culture and capabilities of the organization itself. They also do little to isolate critical components of a project that contain the highest degree of risk. To quantify the potential risks of a new project (or even a current project), we must turn to the history of project management in the organization's past.. The use of a Case-Based Reasoning (CBR) tool to evaluate the structure and resource requirements of new projects based on the degrees of success or failure for similar projects in the past gives management a powerful tool to forecast problems and allocate resources. Because CBR systems absorb current projects into their case base, they also provide a continual method of refining and adapting their estimates.

For more information or to schedule a presentation call (919) 678-0477 or visit www.scianta.com

A Project Risk Assessment Model



Scianta Intelligence

©2004 Scianta Intelligence, LLC
AR-PA-005