Ali Almossawi <almossawi@cmu.edu> / <almossawi@alumni.cmu.edu>
August 13, 2006

Automating the Human-Involved Parts of CURE

Abstract

In evaluation methods that cost tens of thousands of dollars or more, any amount of automation can contribute to significantly reducing cost. When it so happens that such methods require human judgment, the problem of automation becomes more challenging. This document first discusses, briefly, how the COTS Usage Risk Evaluation (CURE) method was partially automated with the help of a tool and then proposes a way of automating its most human-involved part by coding choices beforehand and then executing resolution strategies for different cases.

1. Introduction

The COTS Usage Risk Evaluation (CURE) method is a method that takes five days to complete and has a price tag in the order of tens of thousands of dollars. Its results are extremely useful for organizations that need to know whether or not there is any risk associated with their usage of one or more COTS products. In order to help reduce the cost of CURE, a tool was created to partially automate it. The possibility of full automation was then investigated; the results of that investigation are discussed in this document in the form of a proposed solution.

Section 1 gives some background information on CURE and the CURE process. Section 2 describes the tool that was created to partially automate CURE, which aids the evaluators during the entire interview stage and automates the conversion shown in the first box (see figure 1) of the analysis stage. The consolidation of differently instantiated risk factors that follows is done manually as per the client's request. Section 3 proposes a way of automating the consolidation part. The conversions that occur thereafter are for the most part results of executing database queries, so not much effort would be required to automate things there.

Decision making and consensus models were investigated as well as their related fields to try and see how best to automate processes that include some amount of human involvement. An initial thought was to mimic real life by having discussion and decisions stages and facilitators and evaluators as actors. Each evaluator was foreseen as having an attribute that kept track of his[1] energy level, which would consequently determine his inclination to argue in cases where there would be disagreements. The current solution is much simpler; it classifies all the possible ways that the evaluators can act during the analysis stage and then associates each of those classes with a resolution strategy. The possible choices of all the risk factors are also coded beforehand to limit the likelihood of multiple interpretations of answers later on.

1.1. Definitions

- CURE document: A document containing chapters, discussion topics and risk factors that an evaluator uses to ask questions and record responses during an interview[2].

[1] "his or her" throughout

[2] For further information about the structure of a CURE document, the reader is kindly referred to [1].

- Chapter: A collection of sections in a CURE document.

- Section: A collection of discussion topics and risk factors in a CURE document.

- Discussion topic: An itemized list of information covering a particular issue that an evaluator uses to form his question. Discussion topics have one-to-one mappings to risk factors.

- Risk factor: A statement with one or more blank spaces within it that effectively constitutes the answer to a discussion topic. When a risk factor's spaces are all filled in, it is described as being instantiated.

- Harmonization: The process of merging three CURE documents together such that the risk factors that were instantiated with the same answers are hidden from view and those that were instantiated with different answers are displayed.

- Consolidation: The process of resolving all disagreements in a harmonized CURE document with respect to differently instantiated risk factors, which culminates with a single CURE document.

1.2. About CURE

CURE was developed at the Software Engineering Institute (SEI) as a means of informing organizations of whether or not there is any risk associated with their usage of some number of COTS products. Questions such as how long a product has been in the market and how well it is supported are two of over 300 questions that CURE evaluators elicit answers to. As a result of bringing such questions to the surface, the evaluators can then make a judgment about the amount of risk associated with an organization adopting a particular COTS product [1].

CURE is split into two stages: an interview stage and an analysis stage. In the interview stage, three evaluators go to an organization and sit down with a representative who has intricate knowledge of the project at hand. All of the evaluators have individual copies of the CURE document in front of them and so they spend the interview independently jotting down notes risk factors. One of the evaluators, the lead questioner, is responsible for asking all the questions.

The analysis stage starts the next day. The evaluators sit around a table and begin the process of consolidating risk factors that they disagreed on. Once all the risk factors have been instantiated, they input them into a database and, following a number of manipulations, convert them into risk conditions. The risk conditions are then, effectively, converted a few more times until they're in a form that allows one to see what the risks, strengths and mitigations are for the COTS products that are being evaluated. The team then makes a decision for each of those COTS products. So for example, they might conclude that there are no risks associated with product x because of some number of strengths, but there *are* risks associated with product y because of some number of reasons.

2. Partial Automation

Similar to many of the other methods developed at the SEI, CURE relies heavily on human involvement. The aim of the devised tool isn't to completely remove the human from the loop, but rather to provide a tool that supports his current activities. CURE takes five days: one day for the interview and four days for the analysis. One of the tool's high-level goals was to reduce that by one day, thus resulting in a 20% improvement. CURE's developers have already said that they foresee the tool saving them thousands of dollars. Exactly how much time it cuts off from an actual CURE has yet to be discovered.

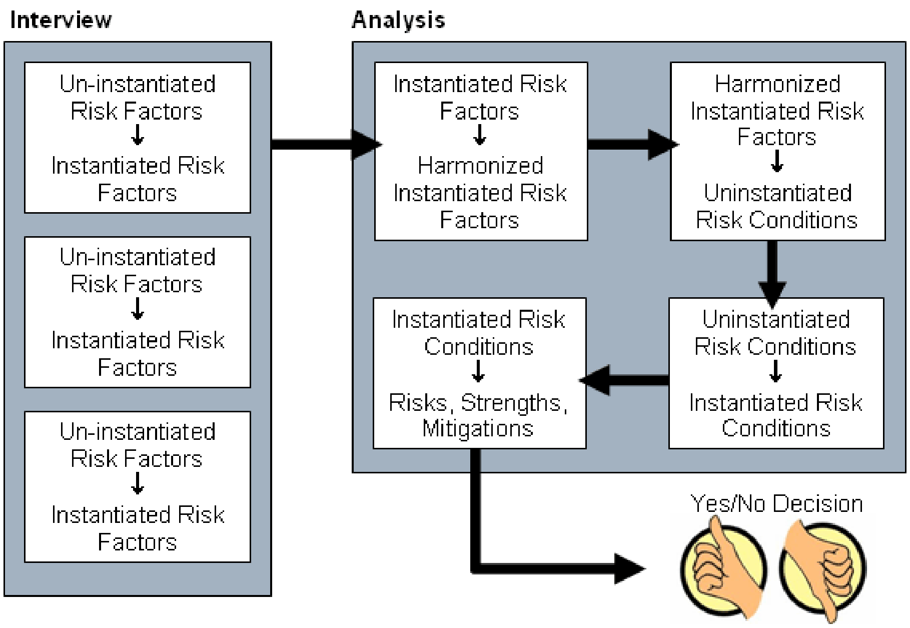

Figure 1: CURE's parts. The tool automates the conversion shown in the first box of the analysis stage and aids the evaluators in completing the conversion shown in the second box.

The tool displays the entire CURE document when the interview mode is set and allows the evaluator to page through it, instantiating risk factors and jotting down notes for them as desired. The electronic version of the CURE document completely replaces the paper version of it. Once the interview is over, the mode is switched to analysis, at which point two of the evaluators export their projects and import them into the copy of the tool running on the third evaluator's machine. Pressing a button then harmonizes the three documents by showing the risk factors that were instantiated differently in a table. Upon clicking on each table entry, the evaluators can see the differently instantiated risk factors and mark one of them as the agreed-upon final version. When no more risk factors remain, the resulting CURE document is said to be consolidated.

3. Full Automation

3.1. Groundwork

The decision table model is used as a basis for the decision-making process. For any given risk factor, a simple table is used to decide what action to take based on how the evaluators answered it. The actions are based on the scenarios that can occur in real-life under different situations. These scenarios are:

- The evaluators all agree on an answer and thus proceed to instantiate that risk factor.

- An evaluator points out to one or both of the other evaluators an observation that he or she made that the others missed.

- Two evaluators agree on an answer, but the third disagrees; the latter is simply too tired to argue so he goes along with their judgment.

- All of the evaluators made observations that are valid, but perhaps subtly different. Their discussion goes on for a fairly long time before one of them caves in (probably from exhaustion) and they finally all agree on an answer.

As is evident from the above scenarios, there is some degree of fuzziness associated with how certain disagreements are resolved. This becomes more apparent as the analysis stage rolls into its last day and last hours and the team becomes exhausted and thus less inclined to debate much. The fuzziness is taken into consideration and is part of the reason why one of the resolution strategies described in section 3.2 simply considers the answer of the majority to be the correct answer.

Heuristics are used exclusively for the set of rules that determine the validity of evaluators' rationales. So for example, if an evaluator's rationale has a point that says that he decided that the answer for a risk factor is that some COTS product is well supported because its vendor is Oracle, then there may be a heuristic that says that if the vendor of a product is ever Microsoft, Oracle or Sybase then that product is well supported.

3.2. Proposed Solution

There are two concepts that are important in the proposed solution. The first is that all risk factors' choices are coded beforehand to prevent ambiguity later on and reduce the number of potential disagreements. The codes are predetermined, but may be modified by the user. For example, a risk factor may look like the following:

Some organization has
<**No** / **Limited** / **Moderate** / **High**>
experience in modifying {product x}

Now, the lead questioner doesn't read that out during the interview, but instead asks: "How experienced would you say you are in modifying product x?" The interviewee may respond with "We've been doing it for about a month." Each evaluator then either writes down "one month" or goes ahead and instantiates the risk factor based on what he internally translates the value of one month to. In the case of disagreement, there is a good possibility that the evaluators will end up arguing during the analysis stage about whether one month constitutes, say, limited or moderate experience.

Risk Factor 123 {
No = 0
Limited = 1 to 3 months
Moderate = 4 to 12 months
High = More than 12 months
}

What this solution proposes is that all choices be associated with particular ranges beforehand. Given the CURE documents of the three evaluators, the system would then easily be able to use their noted down answers of *one month* to instantiate the risk factor. The requirement here of course is that the evaluators write down the response they hear from the interviewee instead of going ahead and instantiating the risk factor *during* the interview. The system needs to work with the interviewee's response and not what the evaluators thought that response translated to. The biggest problem with the current method in CURE is non-determinism because of the sole reliance on the evaluators' judgments. With the proposed system, this can never be an issue. Mappings may be arguable, but at least they are consistent.

The second concept is the system's logic block, which is in turn made up of two parts: a decision part and a resolution part. Drawing from the decision scenarios that were enumerated in section 3.1, the system considers three possible cases that can occur when the evaluators are faced with a risk factor: 1) They can all agree on the answer, 2) two can agree and one can disagree or 3) they can all disagree. Each of these cases has a resolution strategy associated with it that takes care of resolving that case and thus instantiating some risk factor.

Figure 2 shows a graphical representation of the system. *YYY* means that all the evaluators have the same answer for a given risk factor. *NNN* means that each evaluator has a different answer. And *YYN* or *NNY* in whatever order means that two evaluators have the same answer, but the third has a different one.

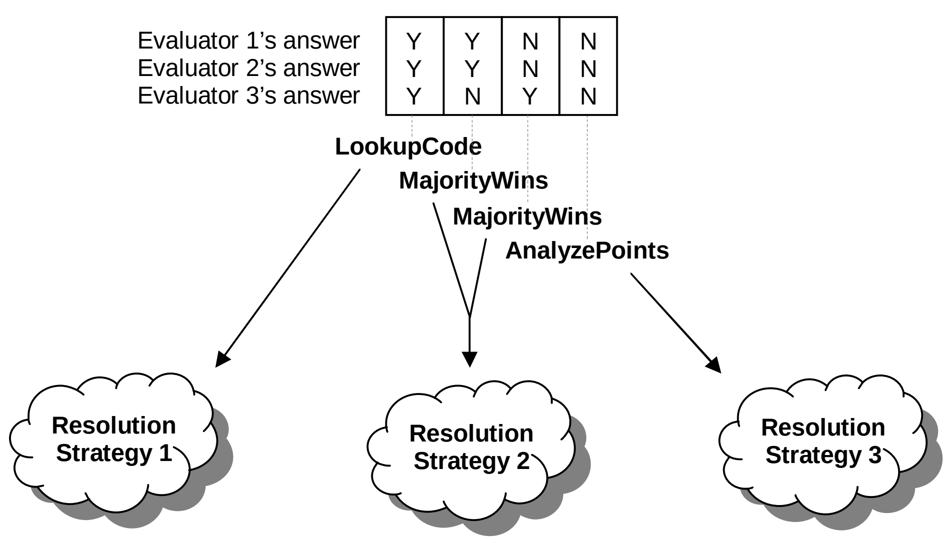

Figure 2: Graphical representation of the proposed solution

In the first case, the *LookupCode* resolution strategy is executed. Since risk factors' choices are all coded, when the evaluators agree on the answer, instantiating the risk factor can be done by simply looking up that answer's code. Although it is acknowledged that consensus isn't an indication of correctness, no attempt is currently made to scrutinize the agreed-upon answer or consider other possible choices.

In the second case, the *MajorityWins* resolution strategy is executed. The assumption here is that because the evaluators have been on so many CUREs together and are trustful of each other, whenever there is a case of disagreement where there is a majority, the majority is assumed to have the correct answer. Some amount of justification exists for how this resolution strategy currently executes, but admittedly, it may be the lazy man's solution. A more clever system may wish to scrutinize the evaluators' answers, especially those of minorities.

The third case's resolution strategy, *AnalyzePoints*, is a more interesting one. Here, because there is no majority to fall back on, some extra work has to be done to resolve the disagreement. What happens here is that the rationale of each evaluator is analyzed and checked against a set of heuristics[3]. Each heuristic has a weight associated with it that represents its value. Rationale entries are individually checked against the set of heuristics and whenever a match is detected, the heuristic's weight is added to the evaluator's internal counter. At the end of the pass, the evaluator with the most points wins and is deemed the one with the correct answer.

The following is an example of how this would work: Consider a case where the interviewee is asked the question: "How well would you say product x is supported?" The response to that may be "Well, I think it's well supported because it's developed by vendor y." One evaluator might interpret that to mean that the product isn't well supported because the interviewee was unsure. The second evaluator might interpret it to mean that it *is* well supported because, after all, it's developed by vendor y. Say the third evaluator interprets it as something else. Now assume there's a heuristic that says that if a product is ever developed by vendors y, y1 or y2 then it's considered to be well supported. In this case, the second evaluator would end up with the most points at the end of the computation and his answer would be considered the right one.

3.3. Measuring Success

The success of the system is measured through empirical means. Ideally, the CURE evaluators would have to observe how the system's results compare with those of previous CUREs. The metric for success is the percentage of similarity between the risk factors instantiated by the system and those instantiated by the evaluators during those CUREs, meaning the number of risk factors that are instantiated in the same way. The assumption is that the previously done CURE that is used for testing constitutes 100% success. A success threshold needs to be determined. At present, an arbitrarily chosen one of 75% has been set.

4. Limitations and Future Work

The solution proposed in this document takes shortcuts such as in cases where there is a majority. Also, when all evaluators agree on an answer, it trusts the result of the consensus and doesn't attempt to look at other possible choices. Further scrutiny of the set of all answers may prove to be more beneficial. The solution needs to be implemented and tested to truly reveal how useful, reliable and consistent its results are. Thereafter, it would be easier to describe some of the concepts discussed herein at a finer level of granularity.

[3] It is yet to be decided how rationales are captured and expressed

5. Acknowledgements

This work has benefited from the comments of Shang-Wen Cheng as well as the suggestions of Gil Taran, who has been an involved mentor for the duration of the project.

6. References

[1] Identifying Commercial Off-the-Shelf (COTS) Product Risks:
The COTS Usage Risk Evaluation, David Carney, Patrick Place and Edwin Morris,
http://www.sei.cmu.edu/publications/documents/03.reports/03tr023.html

[2] Patterns of Problem Solving, Moshe Rubinstein, Prentice Hall, 1975

[3] Cognition: The Thinking Animal (2nd ed.), Daniel T. Willingham, Prentice Hall, 2003

[4] Architecture-based Self-Adaptation in the Presence of Multiple Objectives, Shang-Wen Cheng, David Garlan, Bradley Schmerl, 2006

[5] Decision Tree Primer, http://www.public.asu.edu/~kirkwood/DAStuff/decisiontrees/index.html

[6] The Consensus Decision Process in Cohousing, http://www.cohousing.ca/consensus.htm