

Using Rationale to Assist Student Cognitive and Intellectual Development
Position Paper for the Workshop on Creativity and Rationale in Software Design
Janet E. Burge, Miami University

1. Software, Creativity, and Intellectual Development

Software development is, at its core, a creative enterprise. Given a problem, there are many possible solutions. For some, this is what attracts them to the field—software development as an exercise in creative design. For others, especially as college students, the multitude of solutions, where there is often no clear “right” answer can be a source of frustration. With the many demands on their time, both curricular and extra-curricular, there is significant pressure to find the, or a, correct solution in as little time as possible. The skill of being able to understand just enough about the material to come up with an answer serves them well in some of their earlier courses, where a program is correct if it produces the correct set of outputs given a set of inputs, but they run into difficulty in their later courses where solutions need to be analyzed on multiple dimensions. These difficulties are exacerbated in courses such as Software Engineering and Human Computer Interaction where the system design is influenced not only by the technology available but by how people intend to use it.

Part of the difficulty lies in the need for students to progress through different levels of development cognitively and intellectually before they can tackle creative problem solving. Bloom’s Taxonomy (Bloom, 1956) identifies six levels of student cognitive development: knowledge, comprehension, application, analysis, synthesis, and evaluation. Research has also shown that students pass through several levels of intellectual development. Perry (1970) identified nine positions of development starting with duality, where answers exist for everything and where they can be right or wrong, continuing through several levels of commitment, where students can begin to integrate knowledge and make their own choices based on that information. Moving beyond the search for the fastest correct answer requires that students begin to think at the higher cognitive levels in order to analyze the problem and their solutions, synthesize alternative solutions, and evaluate those solutions by multiple criteria. It also requires that they achieve a level of intellectual development that goes beyond duality and allows them to look beyond their first interpretation of the problem and solution and understand that there is more than one approach and often more than one viable solution. When students are “stuck” at the lower levels of cognitive development, the solution chosen is likely to be routine and familiar, rather than innovative and creative.

It is essential that Computer Scientists, and Computer Science students, think creatively in order to successfully develop software. Glass (1995) describes several aspects of software development where creativity is critical: determining how to translate the customer/business needs into a problem that the software can solve; resolving stakeholder conflicts; designing solutions to new and complex problems; determining test cases; and enhancing existing systems to meet needs that were not initially anticipated by the customer or the developers. A student convinced of a single right answer is likely to either insist that the stakeholder(s) provide this answer (when the stakeholders may not be approaching the problem with an awareness of what is possible with the technology available) or insist that their solution is the only one, or the best one, even if it may not be acceptable to the client.

2. Reflection and Rationale

Reflection serves an important purpose in both education and in practice. Dewey (1933) defined reflective thinking as “active, persistent, and careful consideration of any belief or supposed form of knowledge in the light of the grounds that support it and the further conclusions to which it tends.”

Reflection guides the learning process as evidence is examined and conclusions drawn. Dewey's claim that reflective thinking is necessary when it is not possible to come up with "certain solutions" was the reason why King and Kitchener chose reflection as the basis for their model of student epistemological development (King and Kitchener, 2002). The Reflective Judgment Model (King and Kitchener, 1994) defines seven stages of student epistemological development, broken into three categories: pre-reflective thinking, quasi-reflective thinking, and reflective thinking.

Schön, in his book "The Reflective Practitioner" (1983) describes the need for professionals to move beyond Technical Rationality, where problem solving is the application of theory, to processes that allow for uncertainty and conflict. He describes "knowing-in-action" – where practitioners act based on tacit knowledge, and "reflection-in-action" where practitioners reflect on what they are doing as they do it. Fischer, et al. (1991) describe how design rationale supports reflection by capturing the designer's knowledge about the situation.

Argumentation-based rationale provides a natural mechanism for representing problems, candidate solutions, criteria, and arguments relating those criteria to the candidate solutions. The ability to identify problems, solutions, criteria, arguments, and to apply that information to making a decision under circumstances that involve a certain degree of uncertainty would require that students operate at the higher levels of cognitive development as defined by Bloom's Taxonomy as well as moving past the duality stage into higher levels of intellectual development that allow them to commit to an alternative when the "best" solution is not necessarily clear.

3. Rationale and Creativity

There are a number of views of the creative process. One is to view creativity as a balance between divergent thinking, where many alternatives are generated, and convergent thinking where one option is chosen and explored in more detail (Stoycheva and Lubart, 2001). Rationale supports both the capture of the alternatives and their exploration, by supporting the evaluation of the more promising alternatives and any additional decisions required during their elaboration. Rationale also can play a key role in explaining a creative solution that, because it is a new approach to the problem, may not be as easily understood as a more routine or familiar one would be.

Problem-solving can be broken into four stages: problem definition and analysis; idea generation; idea evaluation and selection; and implementation of the selected idea (VanGundy, 1981). Rationale can support some idea generation techniques, such as Brainstorming, by representing alternatives as generated, and Attribute Listing, a technique developed by Crawford (VanGundy, 1981), where attributes listed would be alternatives. Rationale captured in the form of argumentation is especially useful, however, during the evaluation and selection stage by capturing criteria, their relationship to the alternatives, and supporting evaluation. Some of the techniques described by VanGundy (1981) that could be supported by rationale are the Advantage-Disadvantage approach, enumerating the advantages/disadvantages of each alternative with respect to a pre-defined set of criteria; the Battelle Method (Hamilton, 1974; VanGundy, 1981), dividing criteria into culling, rating, and scoring in order to narrow the field of alternatives; and Reverse Brainstorming (Whiting, 1958; VanGundy, 1981) (brainstorming on the disadvantages of each alternative). Rationale systems that perform evaluation, such as the Software Engineering Using RATIONale (SEURAT) system (Burge and Brown, 2004), can be considered a type of Weighting System (VanGundy, 1981) by allowing weights to be assigned to the criteria and using those weights in evaluation.

Lateral thinking (de Bono, 1970) is related to creativity. Lateral Thinking, like many of the problem solving techniques, involves generating alternatives. Lateral Thinking also requires capturing and challenging assumptions. If a rationale-based approach is to support Lateral Thinking, it also needs to be

able to capture assumptions so that they can either be considered or challenged. VanGundy's (1981) comparison of Lateral and Vertical Thinking shows that one of the key differences between them is that Vertical Thinking tries to find absolutes, yes/no answers, and a "right" solution, while Lateral Thinking is interested in new ideas and resists ruling out "irrelevant" information prematurely. This relates to the levels of student intellectual development described earlier and suggests that in order to think laterally, students need to have progressed beyond dualism.

4. Software Engineering Using RATIONale

Teaching critical thinking, reflection, and idea exploration can be supported using a Rationale Management System to capture and evaluate the alternatives generated by the students. One example is the Software Engineering Using RATIONale (SEURAT) system (Burge and Brown, 2004). SEURAT captures rationale as structured argumentation (decision problems, decision alternatives, and arguments) and uses both the structure of the rationale (syntax) and the content (semantics) to inference over the rationale to detect incompleteness (of the rationale) and inconsistency (of the design). The arguments in SEURAT can refer to system requirements, desired qualities, assumptions made, and relationships between alternatives. Figure 1 shows some rationale as captured in the SEURAT Rationale Explorer. SEURAT stores the rationale in a relational database, allowing the rationale to be shared between multiple users during collaborative decision-making.

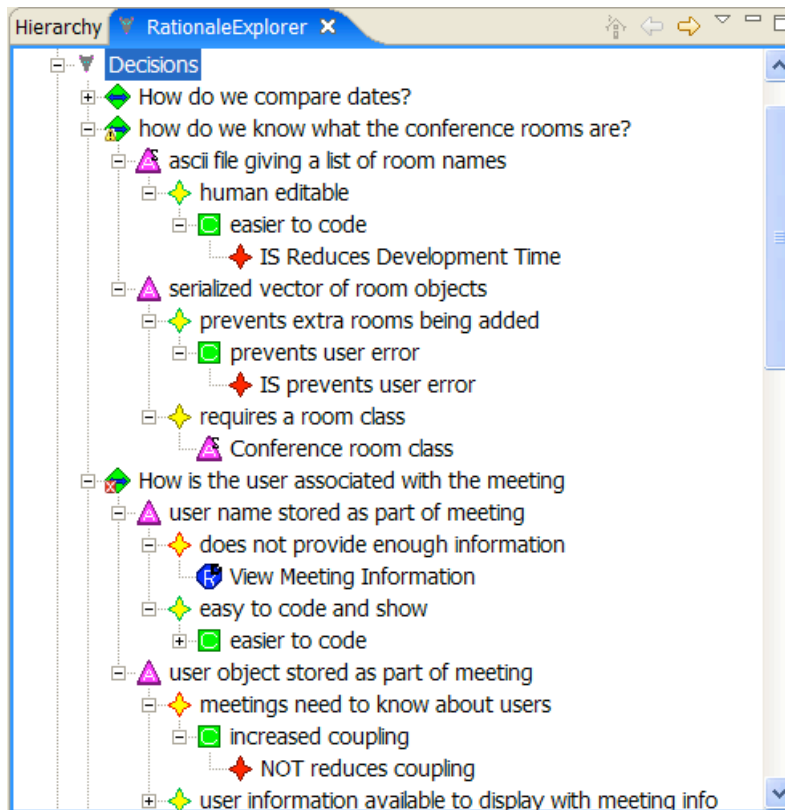


Figure 1. SEURAT Rationale Explorer

The above example shows three decisions, taken from the rationale for a Conference Room Scheduling System. The decisions are displayed using a diamond-shaped icon containing a double-headed arrow. The second decision, "how do we know what the conference rooms are?" has a warning icon overlaid on

it. This is because the alternative selected, “ascii file giving a list of room names”, is not as well supported as the other candidate alternative, “serialized vector of room objects.” The third decision, “How is the user associated with the meeting,” has an error icon because none of the proposed alternatives has been selected yet.

The semantic inference performed by SEURAT is supported by an Argument Ontology of common reasons (criteria) for making decisions. These reasons can be given a priority specific to the system being developed. The Argument Ontology terms (also known as Ontology Entries) can be mapped to claims that an alternative does, or does not, support that criteria. Claims can be re-used throughout a system and either inherit their importance from the ontology or be given a different one. Arguments for and against decisions can then refer to these claims (once again having the option of inheriting or changing the priority). Uncertainty in decision-making can be captured in the rationale in a number of ways. Each argument can have a plausibility associated with it to capture how certain the designer is about its validity; arguments can refer to assumptions, capturing cases where the criteria may eventually no longer hold true; and designers can also capture any questions that need to be answered before finalizing their decisions. The ability to capture uncertainty can assist with the creative process by allowing the decision-maker to capture ambiguity and then move on without needing to fully solve the problem.

5. Using Rationale in Software Engineering Education

There are a number of ways that rationale could be used to support student learning. At the earlier stages of their education, students can be presented with the problems, candidate solutions, and the rationale. For example, in a data structures course the students learn many different ways to represent collections of objects. The student focus is often on how to implement these collections. While the implementation is certainly important, since many of these constructs are often supplied with the programming language (and do not require implementation) it is often more important that the students understand why they might choose a particular data structure for a problem, i.e. to *analyze* the possible solutions by determining which criteria are relevant to the specific problem. The students’ emphasis on the implementation rather than the selection becomes apparent in later classes where they tend to stick to one or two favorite structures which may or may not be the best choices for the problem at hand. Providing rationale in a form that can be easily understood and manipulated may be a more effective way to teach the students the tradeoffs involved in selecting between data structures. The ability to manipulate the argument criteria can also help the students to explore how changing priorities result in different preferred solutions. Rationale can be presented in a form where the criteria can be manipulated by modifying their relative importance in order to demonstrate how as criteria change, so should the recommended solution.

When the students are comfortable with the idea of multiple alternative solutions, the next step would be to involve them in exercises where the problems and criteria are provided but where the students need to identify (*synthesize*) the candidate alternatives based on what they have learned in class and on their own experience. An example of this would be if students were asked to provide alternative methods for data entry or visualization based on usability criteria that they have learned in an HCI course.

The ability to identify the problems themselves, propose solutions, and define criteria requires *evaluation*—identifying what aspects of the solutions are important to the problem and its context. This is an essential skill in both software requirements analysis and in design. The requirements elicitation process is one of defining the problem and the criteria under which the solution will be evaluated while the design process involves identifying and selecting solutions to that problem.

The movement from dualism through multiplicity and into relativism and commitment is more of a challenge. Kloss (1994) recommends several strategies to move students from dualism towards relativism that stress the importance of analyzing and structuring different points of view. This requires looking at the alternatives and evidence, including understanding the role of assumptions. As students move from working with rationale of others to producing rationale themselves, the rationale can serve as both an instructional tool and as a means of assessing their intellectual development.

Table 1 lists the levels of the Bloom Taxonomy, how the Reflection and Rationale approach should support those levels, and how students at different levels of development, as measured by the Perry Scale, would perform on the rationale-supported tasks.

Table 1: Relating Bloom’s Taxonomy, Reflection and Rationale, and the Perry Scale

Bloom	Reflection and Rationale	Perry
Knowledge	Given a general decision problem, list and define the alternative solutions, as described in class. Ex: what data structures can be used to store lists of items?	Students at all levels should be able to do this since it should be directly recalled from their lecture notes.
Comprehension	Given a general decision problem, and a set of criteria for making a selection, explain why these criteria are important. Ex: why is it important to be able to efficiently remove elements from a list of items? Given a set of alternatives for a general decision problem, differentiate between them (this may require giving the students the criteria). Ex: what is the difference between two data structures that store lists of items?	Students at all levels should be able to do explain the criteria but students still in Dualism may show biases toward certain alternatives (the “right” one) when differentiating between them and may not explore them in detail.
Application	Given a specific decision problem, give a list of possible solutions. Ex: give a specific design problem (perhaps one that the students were doing as a class project and are familiar with) and ask which data structures could be used to solve it	Students in Dualism may have difficulty providing more than one solution or more than one valid solution.
Analysis	Given a specific design problem, give a list of possible solutions and map those to a set of design criteria. Ex: given a specific design problem, list the appropriate data structures and how they relate to a set of provided criteria (such as access time, memory use, etc.)	Students in Dualism may have difficulty providing more than one solution or more than one valid solution.
Synthesis	Given a specific design problem, define the criteria that should be used in order	This is not clear. Will students in Dualism only come up with criteria that apply to

	to make a decision. Ex: given a specific design problem, what criteria are important in choosing a solution?	their chosen alternative? Will students in multiplicity have issues of importance?
Evaluation	Given a specific design problem, define alternatives and the key criteria, and use this information to select a solution. Ex: given a problem, and a set of functional and non-functional requirements for the system it applies to, what are the alternative solutions, criteria that differentiate them, and the relationships of these criteria to the requirements? Based on these criteria, which alternative should be selected as the solution?	Students in Dualism may have difficulty defining multiple alternatives; students in Multiplicity may have difficulty making a selection.

Appendix A describes a set of experiment objectives and the metrics used to determine the success of rationale at aiding and evaluating student cognitive development.

6. Related Work

The Moran and Carroll (1996) book included two approaches to using rationale in education. The first was to provide rationale in the form of templates to assist with UI design (Casaday, 1996). The templates help the designers to “ask the right questions” and assist the designer with the process by guiding them toward a solution. Carey et al. (1996) built a library of “exemplary user-interface designs” along with their rationale so those examples could be used to teach HCI design. The work proposed here will use some of the ideas of this work but will also support additional manipulation and evaluation of design criteria, including assumptions, as well as using rationale to assist with the definition and documentation of new designs.

7. Initial Results

Two informal experiments have been performed using rationale in an educational setting. Both these experiments involved using the Software Engineering Using RATIONale (SEURAT) system (Burge and Brown, 2004) to capture and analyze the rationale. The first experiment was conducted during a training exercise for new college hires at a small consulting firm. Observations of the instructor (an employee of the consulting firm) were as follows:

Everyone said SEURAT did not help much but would be useful if it were integrated into a design tool. They view Eclipse as being too late in the process. However, I noticed that the teams that made a real effort to record decisions in SEURAT also were the ones with the most concrete discussion of design and analysis. It wasn't my intent but one team was much more superficial in their design than the other two. They also said they did not use SEURAT very much. The other two did try to use SEURAT and even though they did not see a value, their designs were far more complete and clearly they engaged in a more thorough process and lively discussion. I can't say there is a cause and effect here but this limited data does seem to support some correlation.

The second experiment involved a laboratory exercise about rationale that was administered during two semesters as part of a software engineering course. The students were given a set of requirements and decisions. Some alternatives were provided while others were left for the students to decide. During the first semester, eight out of fourteen students either agreed or strongly agreed that explicitly identifying alternatives and capturing the rationale gave them a better understanding of the specific design decisions addressed in the lab, with the remaining six being undecided. During the second semester, sixteen out of seventeen students agreed or strongly agreed with the above statement with one disagreeing. No student groups in the first semester were able to complete the lab exercise in the time allowed so that may have been a factor in the large number of undecided responses.

Neither of these experiments yielded definitive results. In the consulting firm training experiment, the instructor's observation about cause and effect is correct—with only observing three teams there is no way to determine if the teams used SEURAT because they were the most diligent about exploring the problem or if the use of SEURAT and rationale resulted in a more concrete discussion and design. With the software engineering laboratory experiment, the large number of undecided responses makes it difficult to assess the impact for the first exercise although the results were very promising from the second. The laboratory exercise will be repeated in future offerings of the software engineering course to collect more data.

8. Summary and Conclusions

Students progress through several stages as they gain in knowledge and experience. They run into difficulty when they need to operate at higher cognitive levels than they are accustomed to. The ability to make decisions when confronted with uncertainty and ambiguity is important as the problems they tackle become more realistic and beyond the point where, if they perform the right sequence of actions, they can produce a single correct answer. The ability to synthesize and evaluate solutions becomes critical for problem solving and creativity. Related to this is the need for students to move beyond duality, where there are always right and wrong answers, towards higher levels of thinking where they can begin to analyze the evidence and understand that not all criteria are equally valid in every context.

Rationale provides both a mechanism to express the results of this analysis, synthesis, and evaluation as well as providing some assistance during the process, especially if it can assist with evaluation. Explicitly expressing rationale for their work encourages both reflection on why they made their choices and the active consideration of multiple alternatives. The resulting rationale also provides insight into their thinking for the instructor to use to assess both the student's understanding of the problem at hand and where they are likely to be in their development along the Perry scale. Understanding where the students are at, developmentally, relative to where we want them to go is important in deciding how to help them progress. Rationale can be a valuable tool in both aiding and assessing that progression.

9. References

- Bloom, BS (ed) (1956) *Taxonomy of Educational Objectives: The Classification of Educational Goals*. David McKay Company, Inc. New York.
- de Bono E (1970) *Lateral thinking: Creativity step by step*. Harper and Row, NY.
- Burge JE, Brown DC (2004) An integrated approach for software design checking using rationale. In: *Design Computing and Cognition '04*, Gero J (ed) Kluwer Academic Publishers, pp 557-576.
- Carey T, McKerlie D (1996) HCI Design Rationales as a Learning Resource. In: *Design Rationale: Concepts, Techniques, and Use*, Moran TP, Carroll JM (Eds). Lawrence Erlbaum Associates. New Jersey. pp 373-392.

Casaday G (1996) Rationale in Practice: Templates for Capturing and Applying Design Experience. In: Design Rationale: Concepts, Techniques, and Use, Moran TP, Carroll JM (Eds). Lawrence Erlbaum Associates. New Jersey. pp 351-372.

Dewey J (1933) How We Think. D.C. Heath and Company: New York.

Fischer G, Lemke AC, McCall R, Morch A (1991) *Making Argumentation Serve Design*, Human Computer Interaction, 6(3-4):393-419.

Glass, RS (1995) Software Creativity, Prentice Hall, New Jersey.

Hamilton, HR (1974) Screening Business Development Opportunities, business Horizons, August, pp. 13-24.

Perry, WG (1970) Forms of Intellectual and Ethical Development In the College Years: A Scheme. Holt, Rinehart, and Winston. New York.

King, PM, Kitchener, KS (1994). Developing Reflective Judgment. San Francisco: Jossey-Bass Publishers, pp. 14-16.

King, PM, Kitchener, KS (2002). The reflective judgment model: Twenty years of research on epistemic cognition. In B. K. Hofer and P. R. Pintrich (Eds.), Personal epistemology: The psychology of beliefs about knowledge and knowing, (pp. 37-61). Mahway, NJ: Lawrence Erlbaum, Publisher.

Kloss, RJ (1994) A Nudge is Best: Helping Students through the Perry Scheme of Intellectual Development. Journal of College Teaching. 42(4):151-158.

Moran TP, Carroll JM (Eds.) (1996) Design Rationale: Concepts, Techniques, and Use. Lawrence Erlbaum Associates. New Jersey.

Schön, DA (1983) The Reflective Practitioner. Basic Books, Inc. New York.

Stoycheva, KG, Lubart, TI (2001) The Nature of Creative Decision Making, In C.M. Allwood and M. Selart (Eds.), Decision Making: Social and Creative Dimensions, Kluwer, pp. 15-33.

VanGundy, AB (1981) Techniques of Structured Problem Solving, Van Nostrand Reinhold Company

Whiting, CS (1958) Creative Thinking, Reinhold, NY.

Appendix A: Experiments Using Rationale to Assist Student Cognitive and Intellectual Development

The effect of rationale on assisting student cognitive development can be investigated using a series of experiments. Ideally, these experiments would be performed with two groups of students – one using SEURAT and rationale and the other not. It is still undecided what needs to be presented as inputs to the students in the control group. In addition to studying the direct results of using the rationale, the students should be given pre and post tests on the subject matter.

Table 2 describes the objectives, assessment techniques, and metrics used during the experiments. Each objective describes a different experiment. The experiment used will depend on the current and target level of student cognitive development (which will be specific to the level of the course being taught). The metrics provided evaluate the rationale provided by the students. We may also want to perform these exercises over time to see if the student progresses.

Table 2: Experiments on the Effect of Rationale on Student Cognitive Development

Objective	Assessment	Metric
Assist students with identifying which criteria are relevant to selecting a solution to a problem	Students are presented with a decision problem, multiple alternatives, and attributes for each of these alternatives. The students are asked to rate the individual attributes in importance relative to the problem. The SEURAT tool uses their ratings to calculate the support for each alternative. The students then select an alternative.	<ul style="list-style-type: none"> • Are the ratings provided by the students correct? • Do the students select an alternative based on the support calculated by SEURAT? • Do the students modify their ratings after SEURAT's initial assessment prior to selecting an alternative (in order to make the one they select have the highest score)?
Assist students with synthesizing alternative solutions to a problem	<p>Students are presented with a decision problem and a set of criteria, with relative weights (importance) provided and asked to produce a set of alternative solutions based on what they have learned in class and assign criteria to each alternative. The students will also be asked to make a selection.</p> <p>After making their initial decision, the weights will be changed.</p>	<ul style="list-style-type: none"> • Are all the relevant alternatives identified and provided by the students? • Are the criteria appropriately mapped to the alternatives? • Do the students select an alternative based on the support calculated by SEURAT? • Do the students modify their criteria assignments after SEURAT's initial assessment but prior to selecting an alternative (in order to make the one they select have the highest score)? • Do the students change

		their decision after the criteria change?
Assist students with developing alternative solutions to a problem and the criteria used to evaluate them	Students are presented with a decision problem and are asked to provide a set of alternatives and the criteria arguing for and against those alternatives. SEURAT is used to provide alternative evaluation using the criteria provided by the students.	<ul style="list-style-type: none"> • Are all the relevant alternatives identified and provided by the students? • Are the arguments for and against those alternatives correct? • Are the arguments complete? • Are the arguments consistent? • Do the students select an alternative based on the support calculated by SEURAT? • Do the students modify their criteria assignments after SEURAT's initial assessment but prior to selecting an alternative (in order to make the one they select have the highest score)?

Evaluating the experiment results involves evaluating the rationale produced. Table 3 elaborates on the metrics used to determine if the objectives have been met.

Table 3: Detailed Evaluation Metrics

Objective	Assessment	Metric	Excellent/High	Good/Medium	Poor/Low
Assist students with identifying which criteria are relevant to selecting a solution to a problem	Students are presented with a decision problem, multiple alternatives, and attributes for each of these alternatives. The students are asked to rate the individual attributes in importance relative to the problem. The SEURAT tool uses their ratings to calculate the support for each	Are the ratings provided by the students correct?	All the ratings provided by the students are correct	A majority of the ratings provided by the student are correct	The student provides incorrect ratings or fails to provide ratings
		Do the students select an alternative based on the support calculated by SEURAT?	The student selects an alternative based on the level of support	The student select some alternatives based on the level of support	The students do not appear to have reasons for making their selection.
		Do the students modify their ratings after SEURAT's initial assessment	The student has initially correct ratings and chooses the best supported alternative OR	The student modifies some ratings to justify the choice they prefer.	The student manipulates many of the ratings to justify the choice they prefer

	alternative. The students then select an alternative.	prior to selecting an alternative (in order to make the one they select have the highest score)?	the student corrects ratings prior to making a decision The student explains their choices.	The student provides some explanation for their choices	The student does not provide explanations for their choices
Assist students with synthesizing alternative solutions to a problem	Students are presented with a decision problem and a set of criteria, with relative weights (importance) provided and asked to produce a set of alternative solutions based on what they have learned in class and assign criteria to each alternative. The students will also be asked to make a selection. After making their initial decision, the weights will be changed.	Are all the relevant alternatives identified and provided by the students?	The student provides all the relevant alternatives	The student provides most of the relevant alternatives	The student only produces one alternative
		Are the criteria appropriately mapped to the alternatives?	The student maps all the criteria to the correct alternatives	The student maps most of the criteria to the correct alternatives	The student does not successfully map criteria to alternatives
		Do the students select an alternative based on the support calculated by SEURAT?	The student selects an alternative based on the level of support	The student select some alternatives based on the level of support	The students do not appear to have reasons for making their selection.
		Do the students modify their criteria assignments after SEURAT's initial assessment but prior to selecting an alternative (in order to make the one they select have the highest score)?	The student has initially correct criteria assignments and chooses the best supported alternative OR the student corrects the criteria assignments prior to making a decision The student explains their choices.	The student modifies some criteria assignments to justify the choice they prefer. The student provides some explanation for their choices	The student manipulates many of the criteria to justify the choice they prefer. The student does not provide explanations for their choices
		Do the students change their decision after the criteria change?	The student looks at differences in support levels and changes their decisions	The student sometimes fails to change their decision but instead stays with their initial plan	The students do not acknowledge the effect of changing criteria

Assist students with developing alternative solutions to a problem and the criteria used to evaluate them	Students are presented with a decision problem and are asked to provide a set of alternatives and the criteria arguing for and against those alternatives. SEURAT is used to provide alternative evaluation using the criteria provided by the students.	Are all the relevant alternatives identified and provided by the students?	The student provides all the relevant alternatives	The student provides most of the relevant alternatives	The student only one (maybe two) alternatives
		Are the arguments for and against those alternatives correct?	All arguments for and against the alternatives provided are correct	Most arguments for and against the alternatives are correct	The student provides several arguments that are incorrect (or fails to provide arguments)
		Are the arguments complete?	The student provides all or most relevant arguments	The student provides most necessary arguments	The student is missing many arguments
		Are the arguments consistent?	The arguments are consistent (criteria do not conflict)	Most arguments are consistent except in a few places	The arguments are not consistent and include numerous contradictions and missing information
		Do the students select an alternative based on the support calculated by SEURAT?	The student uses the support to make their choice	The student uses the support to make most choices	The student ignores the recommendations made by SEURAT
		Do the students modify their criteria assignments after SEURAT's initial assessment but prior to selecting an alternative (in order to make the one they select have the highest score)?	The student has made correct arguments and chooses based on the support level or they make corrections because SEURAT pointed out a problem	The students make some small changes to justify the choice they prefer	The student ignores SEURAT's recommendations (see above) or manipulates most criteria to justify what they want to choose.