Supporting Creativity in Problem Solving Environments

Marc Vass John M. Carroll Clifford A. Shaffer
Department of Computer Science
Virginia Tech
Blacksburg, VA 24061-0106
mvass@csgrad.cs.vt.edu, {carroll, shaffer}@cs.vt.edu

ABSTRACT

We seek to provide a theoretical basis for the development of problem solving environments that support creativity. This paper combines flow theory, the systems model of creativity, and a newly developed workflow of problem solving to produce a theory of the creative problem solving user, WorkFlow. It extends the definition of usability to include creativity and identifies key areas and methods for the support of creativity in problem solving.

Categories & Subject Descriptors: H.5.2: Theory and Methods

General Terms: Theory

Keywords: Problem solving, creativity, creativity support tools, usability

INTRODUCTION

As the use of problem solving environments (PSEs) becomes increasingly important to researchers and scientists [27], the need for a theory on which to build environments that support creativity in problem solving has become evident [26]. Usability has become a concern in the development of PSEs, but in a limited scope. The previous focus on system usability increases user satisfaction and reduces error-rates, yet it does not improve the end product of the user's work. Extending the concept of usability to include creativity in problem solving greatly expands the potential to help users. To be successful we require a theory to support this expansion.

A problem solving environment (PSE) is a computational system that provides a complete and convenient set of high level tools for solving problems from a specific domain. A PSE allows users to define and modify problems, choose solution strategies, interact with and manage appropriate hardware and software resources, visualize and analyze results, and record and coordinate extended problem solving tasks. A user communicates with a PSE in the language of the problem, not in the language of a particular operating system, programming language, or network protocol [23].

The most difficult problems to solve lie in areas that require researchers to engage in collaborative problem solving to create a symbolic model that solves a problem in their domain of interest. A symbolic model codifies the researchers' beliefs about the inner workings of the variables in a problem such that given a set of inputs that define the problem instance, a method or methods can be used to execute the symbolic model, to produce the predicted solution to the problem.

This notion of a symbolic model helps us distinguish between forms of creativity that do and do not involve problem solving [19, 24, 25]. Furthermore, we distinguish between problem solving using an already defined symbolic model (algorithmic) [21], and problem solving that requires the construction of a new symbolic model (theory building) [21]. The first type only requires that the solver apply inputs to the symbolic model, as in the rules of algebra for determining what the value of X is, given that Y=2 and Y=X^4. The second type of problem solving requires the solver to extend a previous model or to create a new symbolic model. For example, the modeling of the cell cycle in fission yeast [27] extends previous models to account for new experimental findings. The first type of problem solving requires repeated evaluation to determine correctness, whereas the second requires a creation and evaluation loop as provided for in the Geneplore model [10]. We also distinguish two end results of creativity in problem solving. One is the solution resulting from the application of inputs to a pre-defined symbolic model, and the second is in the creation of a symbolic model.

The design of PSEs has been studied [12, 23], but the issues of complex model creation and evaluation using these environments have not been viewed and applied as creativity. This paper presents WorkFlow, a theory of the creative problem solving user, and applies this theory to PSEs.

CREATIVITY AND FLOW

The difference between a problem solving user and a creative problem solving user is the presence of flow. Flow is an automatic, effortless, yet highly focused state of consciousness. It is the one aspect in which all creative persons are unanimous [4]. Creativity is more likely to result from flow states [5]. Creative persons love what they do, caused by the reinforcement provided by flow upon curiosity and problem solving [22]. In these individuals this "almost automatic, effortless, yet highly focused state of consciousness" commonly exists during difficult activities that involve a degree of discovery and novelty. Flow exists in the context of the workflow itself, so the workflow is a chief component in maintaining and creating flow for the user [16, 22].

In each process involved in problem solving the user abstracts and transforms information. What solvers seek to do is focus their attention onto the task at hand, which is an ability seen in those regarded as particularly creative. When the solver is allowed to move comfortably and smoothly among the information the chances for creativity are at their highest. Ideally, within a PSE the computer is not a restriction to creativity, or even a mere assistant. It should become a partner in the creative process of

C&C'02, October 14-16, 2002, Loughborough, Leic, United Kingdom.

problem solving. Support for flow is achieved by supporting each of the conversion and modifier processes involved in problem solving as defined in the following section on the workflow model. This happens when distractions are kept to a minimum and the user is in control and guiding the system, not the system guiding the user. By filtering to the most important tasks for the user, and allowing the user to create their own abstractions, the environment becomes an extension of their mind.

The following are nine characteristics of flow [4] and are what we will base our future analysis of support for creativity in a PSE. We argue that the higher the rate at which these characteristics appear in the behavior of a person, the better the PSE is at supporting flow.

Characteristics of a Flow Experience

1. Balance of challenges and skills
2. Immediate feedback to one's actions
3. Clarity of goals
4. Merging of action and awareness
5. Distractions excluded from consciousness
6. No worry of failure
7. The activity becomes autotelic (an end in itself)
8. User's sense of time becomes distorted
9. Self-consciousness disappears

The first three characteristics are structural requirements for flow to occur [3, 4, 16]. The fourth and fifth characteristics are the actual flow state itself, while the sixth through ninth characteristics are the consequences of the flow experience and are mentioned in the section on PSE evaluation as measurable indicators of the flow state.

WORKFLOW MODEL

We model the actions of a creative problem solver with a workflow network. Any problem solver must follow a set of paths through the network. This network is expressed in Figures 1 and 2. Figure 1 models the systems perspective [6], which views creativity is "any act, idea, or product that changes an existing domain, or that transforms an existing domain into a new one" [4]. Producing a novel model that is useful to the domain and field in which it is applied is a creative act. This act cannot be judged on the model alone. The field of which the person or group is a part must value it, and it must be novel. Along each edge (labeled "m" or "c") in Figure 2, the interactions of Figure 1 take place.

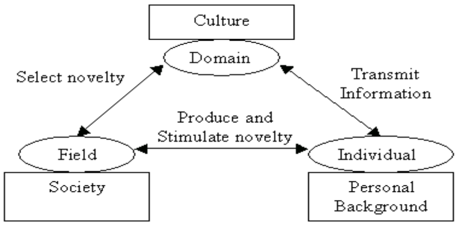

Figure 1. Systems model of creativity.

A path through these figures is defined with respect to all knowledge that is accessible to an individual. A user may begin with a conceptual model produced by another person, but that other person produced their conceptual model by following a set of paths through Figures 1 and 2. Likewise, one can use the results of previous experiments without having performed the experiments personally. Coworkers, referees, and the stored knowledge contained in the domain also have access to the user who might thereby have previously influenced the process. This can occur through comments, suggestions, criticisms, papers, etc. For clarity, these mutual influences are removed from Figure 2; their presence is assumed as a modifier for every conversion process. This "set of paths" concept provides for the likelihood that the processes actually used by an individual may change with time and are not static, even within one instance of a problem.

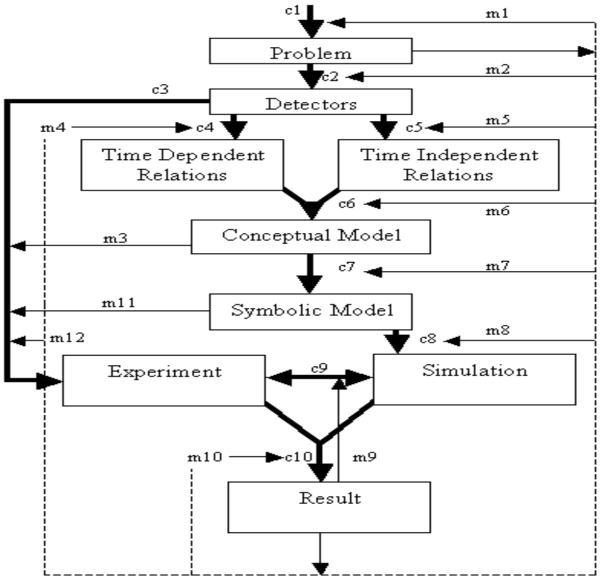

Figure 2. Workflow model of problem solving.

Each arrow labeled with "c" in Figure 2 represents a conversion of information through a process "c" from the information form at the base of the arrow to the information form at the tip of the arrow. These conversions are the primary workflow paths. These conversions and their associated modifiers have been discussed in work from a variety of sources and have been observed in the work of researchers working with the PSE group at Virginia Tech.

Each arrow labeled with "m" represents a modification or influence in process "c" by process "m" which is using the information form at the base of the arrow. These modifiers serve as feedback paths. The primary feedback paths are previous results and problems, reflecting the importance of experience in problem solving. The distinction between modifiers and conversions is made to express that there is no process that can transform an information form into another information form without outside information. For example, a result cannot be transformed into an experiment without some knowledge of what is to be measured in the experiment. Thus, a result can only alter future experiments.

Motivational Problem Solving Scenario

We now provide a motivating scenario for use in discussing the workflow model. Jim is a biologist studying the behavior of a certain mouse species. These mice behave quite oddly in that their wake/sleep schedule is not affected by changes in their daily schedule of artificial light simulating the sun. In other words, they do not experience jet lag. Jim now seeks to explain their behavior by creating a computational model that explains the new mice species' behavior, as well as the behavior of normal mice.

Definitions

We now define the terminology we use to explain the workflow network and scenario.

- A *field* is a topic, subject, or area of academic interest or specialization that contains experts who recognize and validate innovation [6]. In Jim's case, this is the field of molecular biology.
- A *domain* is the activity and works of a sphere of activity, concern, or function. The domain for Jim is the works of molecular biologists.
- A c*ulture* is intellectual activity and the works produced by it [8]. Jim lives in a Western European culture.
- A *problem* is a question to be considered, solved, or answered [8]. Jim's problem is to explain the wake/sleep schedule in two different types of mice.
- A *detector* [17] is a representation of a feature to be observed in the problem. It consists of a binary detector and any number of property detectors. A binary detector is "on" if that feature is present in the problem and is "off" if that feature is not present in the problem. Property detectors represent properties of the feature that are observed. Jim believes that there are a particular group of proteins in mice cells that regulate their wake/sleep schedule. One group of detectors indicate the presence or absence of proteins he believes regulate the mice's schedule. Other detectors are the actual behavior of the mice, such as when they wake up and when they sleep.
- A *time dependent relation* is a relationship between two or more detectors, where time is a factor in the relationship. Jim believes that many of the proteins interact with each other over the course of the day to create the wake/sleep cycle.
- A *time independent relation* is a relationship between two or more detectors, where time is not a factor in the relationship. Jim thinks that the total amount of these mouse proteins remains constant throughout the day.
- A *conceptual model* is a combination of time independent and time dependent relations into a set that contains those relations believed by the problem solver to be useful in solving the problem. Jim's conceptual model is a diagram he has drawn of the interactions between various proteins.
- A *symbolic model* is a conversion of the conceptual model into a declarative, functional, constraint, spatial, or multi-model for determining the state of detectors given a set of inputs [11]. Jim uses a constraint model consisting of differential equations.
- An *experiment* is the specification of a process that in the real world records the state of a subset of detectors over a fixed time period and the conditions under which the process is executed. Jim does not perform experiments. He reads research publications to incorporate the experiments of others into his model.
- A *simulation* is the specification of the methods and inputs used in executing the symbolic model. The inputs for Jim are the types of proteins present in the mouse as well as the rates at which his reactions occur. He executes models using the equation solver LSODAR with a certain group of settings.
- A *result* is a set of qualitative or quantitative evaluations of detector states taken from the union of the detector states produced by a set of executed experiments and the predicted detector states produced by a set of executed simulations. Jim's comparison determines whether the executed simulation matches the behavior of mice in actual experiments.

THE TEN CONVERSIONS AND THEIR APPLICATIONS

We now discuss the conversions that serve as the primary workflow paths and present their applications in a PSE. The scenario is used to illustrate the activities associated with the conversions.

C1 – Problem Formation

The formation of problems is an important field unto itself [1, 13, 25]. Often, the statement of the problem is the key to solving the problem. WorkFlow applies to a problem domain that has already specified a class of problems. This allows for the input of results of previous problem solving efforts and previous problem instances into the problem formation process. From Figure 1 we see that both information from the problem domain and interaction with the field is needed. The field provides motivation and suggestions for the problem. The domain provides the information required by the user to be able to define the problem. The results of solving or attempting to solve similar problems guides the user in discerning whether a problem is solvable, or practical to solve.

Problem instances should be recorded in the PSE and should be searchable by the user. The user should also be able to view the results associated with those problem instances. From the systems view, the problems should be transmittable to others and support feedback from others. Thus, a standard problem definition format should be developed with the assistance of the field that is targeted by the PSE. The user should be able to organize the problem instances in a manner of their choosing, while the PSE should provide this user specific view and the original view to the other users of the PSE.

In our scenario Jim knows the problem he would like to solve and begins his work by looking in the literature for problems involving sleep/wake cycles. He then investigates their associated successes and failures.

C2 – Detector Creation

Previous results in combination with previous problems, determine whether it is appropriate to use a certain detector for a problem. Previous problems suggest detectors that would work with this problem or have been shown to not work with this problem. The field provides feedback on whether this set of detectors chosen by the solver is practical or not and whether it could possibly work. It also suggests directions in which to go in choosing detectors. The domain provides information on detectors that have been useful in the past and what they were used for in some problems. For instance, one may want to know what role a protein played in models for other problems, whether the problems are overtly similar or not, as in Jim asking the question "Is protein *A* only involved in intracellular signaling?"

Detectors should be tied to the features of the problem and detectors for these features should be recorded in the PSE. Detectors used in cellular models will be concentrations of proteins or concentrations of their complexes, whereas detectors for an aircraft design model will be wing shape, structural materials, etc. From the systems view, the detectors should be transmittable to others and support feedback from others. A standard detector definition format would be useful for achieving this. The user should be given the ability to organize detectors into useful groups. Also, the user should be able to make detectors of use available to their entire research group.

Jim decides that other theorists do not use enough detectors to adequately model mouse sleep cycles, and thus he decides to use experimental findings from molecular biology journals as a basis for the detectors he will use.

C3 – Experiment Formation

Experiments are formed from detectors, with the input of results from previous experiments and the problem itself as guidance. As models are developed, both conceptual and symbolic, they further influence what the experiments are designed to test.

The user should be able to define the methods for performing the experiment as well as the detectors to be used. They should be able to personalize the organization of experiments and to have access to the results and the experiments generated for similar detector sets or problems. They should also be able to use previous conceptual and symbolic models to aid them in the selection of features of the problem to instantiate and to observe. To facilitate this selection and organization, a standard experiment definition format should be developed with the assistance of the field.

Jim does not perform experiments. He instead uses the results of experiments from other researchers and then provides these experimentalists requests to perform new experiments. One request he has is to measure the amount of a certain protein at specific times so that he can verify that his model is correct in predicting a protein will increase in amount as the mouse approaches sleep.

C4 and C5 – Time Dependent and Independent Relation Creation

The creation of relations is guided by the results of previous modeling attempts and by the problem itself [17]. Information from literature and peers also serves to influence what relations are created. Intuition serves as the final ingredient in forming relations, which is a more directed form of trial and error. These relations use the language of the domain, expressing abstract concepts of relation, rather than executable statements.

A user should be able to access the relations defined by others within their domain. They should also be able to consult with others in deciding what relations to use and how to define them. This leads to the suggestion that relations be stored in the PSE and organized into categories relevant to the user or domain, such as relations that are involved in regulating a specific protein. The user should be able to define time dependent and independent relations and to use previous relations from related problems. The success of these relations being applied to previous problems should also be made accessible to the user. The relations themselves should be specified in the language of the domain of the user. The specification of language should not be limited to symbols or text alone. A mixture of the two should be possible within the PSE.

Jim decides to use many of the relations that have been theorized in other's models, and to add several time dependent relations of his own that he believes may explain the mice behavior. These relations describe the characteristics of chemical reactions. One relation he adds is that a protein X forms a dimer, XY, with protein Y at a rate of 1.3 per minute.

C6 – Conceptual Model Formation

The formation of a conceptual model is guided by previous results of relation selection and by the problem at hand. The user must link these relations together and filter out the relations that are irrelevant to the problem. The information contained in the domain serves as a basis for forming conceptual models. Information from the domain and comments from others in the field also influence how the user will choose the relations to keep.

Users should be able to select which relations are to be included in their conceptual model and to organize their conceptual model as they see fit. This model itself should be specified in the language of the user and previous conceptual models should be made available to the user, as well as the conceptual models of other individuals in the domain. The PSE should support sketching so that structuring effects not limit the innovativeness of a model [30]. Sketching is the outlining of a set of concepts without particular regard for absolute correctness. Sketching is needed to support the process of formation, since it requires the consideration of many different combinations and requires a view of the overall concepts, while still being able to drill down deeper into the model where required. The conceptual model should support multiple views so that others in the group are able to view it in the manner that is of the most use to them.

Jim's conceptual model begins as a mental representation of various chemical reactions. He then transfers this model to paper. During this time, he makes many revisions to his model until he is satisfied that it should describe the mechanisms behind the wake/sleep cycle. His diagram now represents chemical reactions involving proteins that regulate the sleep/wake cycle.

C7 – Symbolic Model Formation

The formation of a symbolic model is guided by the results of previous symbolic models' simulations and the problem itself. Five types of models have been identified: declarative, functional, constraint, spatial, and multi-model [11]. The symbolic model represents a concretization of the concepts from the conceptual model, yet it is also a simplification of the conceptual model. Capturing the full scope of a concept may not be possible or practical to the user. The results of previous simplifications and the results found in the domain from others in the field play a major role in determining what simplifications to make. Sharing the symbolic model with others allows the user to receive input and review on the simplifications made and possible errors or additional simplifications that could be made.

The user should be able to specify how this conversion from C4 to C5 takes place and to exactly what form of symbolic model the conversion is made. Support for each of these five models should be present within the system so long as it is computationally feasible or applicable for the domain. The user should be able to see the results of previous generations of symbolic models from similar conceptual models from like problems. The user should also be able to personalize the organization of the symbolic model so that components such as equations, variables, and constants appear in the model where the user's mental model places them.

Jim's symbolic model is a series of differential equations, which are a type of constraint-based model. He creates these by writing equations that explain the changes in the proteins with respect to time as a result of the reactions that formed his conceptual model.

C8 – Simulation Formation

A simulation is formed by providing methods for the execution of the simulation and by providing inputs to the simulation based on the symbolic model, problem, and previous results. The results of other simulations may play into this creation and may guide users in the methods they choose to use for simulation. Also, the problem and comments from the field may influence how the methods of simulation are chosen.

The user should be able to choose the methods for converting their symbolic model into a simulation and to be guided in that process by results from previous simulations of like problems and symbolic models. The user should also be able to personalize the organization of simulations for a particular symbolic model.

Jim forms a simulation by giving a symbolic model and its parameters to a simulator. These parameters are chosen from Jim's experience with simulating similar systems of differential equations.

C9 – Experiment Conversion to and from Simulation

A simulation is converted to an experiment by specifying the conversions between variables and inputs in the simulation to variables and features in the experiment. The mapping is sometimes complicated and may be done in both directions. Experiments are often turned into simulations when there are results from other experimenters that the user is trying to account for in their model. In these cases, the mapping is made difficult by the experiments being converted not measuring variables in the same way as the user's simulation. The results of previous formation, input from peers, and others' experiences also play into this conversion.

The user should be able to define how to convert from a simulation variable to an experimental variable and vice-versa. The user should be able to personalize the organization of experiments created from simulations and simulations created from experiments. This conversion linkage should be maintained so that users do not have to recall how mappings are done from one type of variable to another if it is a commonly used method. The user should also be able to use the results of previous conversions to aid them in the current conversion, so as to be able to extend previous conversions if desired.

Jim has defined several conversions from experimental variables to simulation variables. One of these determines the value of a simulation constant representing the rate of synthesis of a particular protein from the type of food being fed to the mouse in an experiment.

C10 – Result Formation

A result is formed by creating quantitative and qualitative comparisons between the detector states resulting from the executions of experiments and simulations. In some cases, a simulation or an experiment will not exist or be comparable to each other. In these cases, the simulation may be accepted by the user as reality or as the best approximation. Also, it may be compared against the simulations of others or to what the intuition of the user would predict. This process is guided by the problem and by the input of others and their knowledge. The results of previous comparisons also play a role in shaping the comparisons used by the user. The user views the results in many different ways and will often do this before attempting to define a comparison.

The user should be able to define how to compare an experimental result to a simulation result and to also be able to evaluate their results individually. They should be able to specify quantitative and qualitative measures for evaluation and to organize those measures for use in later evaluations. The results of previous evaluations should be available to the user so that they can better define comparisons and decide which comparisons are useful in particular cases of experiments and simulations. The user should have access to the comparisons made by others on similar problems. They should also be able to view the results in many different ways, and to define their own method of viewing.

Jim likes to view his results as plots of wake and sleep times for the mice. He views them by overlaying the experimental results with the simulation results. He has also defined an equation that gives a value that defines how closely two results match.

Annotation

Throughout the previous section, interaction between the individual and the field is mentioned as being common. To support this interaction, annotation of the information produced using the PSE should be supported. Examples of annotation include the user recording what their thoughts and rationale were in developing their model, or to state how they chose to evaluate the accuracy of their model. From the standpoint of the reviewer, it allows them to comment exactly on where they have an issue or comment about the model or evaluation.

Error Correction and Process Description

Linus Pauling liked to say that the route to creativity was having a lot of ideas and discarding the bad ones [20]. Perhaps one of the most important parts of problem solving and creativity is recognizing bad ideas and problems in the context of a model or methodology. Good error detection allows a researcher more time for analysis of their promising ideas. The tendency of users to perform the task that takes the least work to achieve a goal plays a role in how a user corrects their errors. When a problem or error is discovered, most users will take the path of least cognitive distance to find the error. Their shortest path will be followed and, if necessary, increasingly distant paths will be followed until the error is found. Cognitive distance is measured in terms of the conversions between related information forms with fewer conversions being needed for more closely related information forms. The modifiers, represented by "m" in Figure 2 provide for this. If the user sees a familiar error, they are likely to proceed directly to the source of the error, which may not be of the least cognitive distance for most users. This can be observed in the behavior of teachers or teaching assistants who are very familiar with a problem and its solution. However, the path they followed was of the least cognitive distance for them, since they had formed an explicit conversion from the error to the source. This illustrates that cognitive distance may vary for different users and for different problems. This occurs by a user placing more emphasis on the modifiers that serve as inputs into their process, such as when a teacher knows the correct method to solve a problem and the common errors made using that method.

Using the workflow model, there is an opportunity to help users correct their own errors, and to even influence them to make fewer errors. This can be done through increasing the importance of the modifiers in the processes if lack of their consideration is a problem, and to make it easier for the user to access them. If the conversion itself is a source of error, then developers can concentrate on it. This has been applied with success in the JigCell Model Builder [28, 29] by changing the process used by biologists to make the conversion from a diagrammatic model of chemical reactions to a differential equation model. The biologist originally would perform this task by hand with a significant error rate. JigCell changed this process from being mathematically oriented to being reaction centered (closer to the biologist's mental model) with the PSE performing the actual conversions from chemical equations to differential equations.

Due to this process of looping, and repeated conversions, there is the opportunity to perform these operations for the user. Since these loops may be of a fluid nature, where possible the environment should allow the user to program it to perform the processes for the user. For instance, Jim may be able to program the PSE to perform simulations to discover if his model accounts for new experimental findings. In this way the entire environment can serve as an assistant to the processes already in place in the user's work or in their group's work.

PSE EVALUATION AND CLASSIFICATION

Previous evaluations of PSEs have been limited to usability reviews and user testing. While useful for assessing support for a specific workflow, such reviews are not enough to assess whether flow is occurring. If flow is occurring, then support of creativity is present.

Measurement of flow has been used to evaluate the likelihood of users enjoying the use of a computer-mediated environment and the likelihood of its future use by users [16]. These studies and others of flow have primarily used self-report questionnaires targeting the characteristics of flow to assess whether users experienced flow [3, 7, 9, 14, 15, 31]. While individual differences play into whether a user will experience flow, the support for the first three characteristics must be present for this state to exist. The consequences of flow have also been correlated to increased learning [31] and increased exploratory behavior [14, 15, 18], both of which can be assessed by behavioral observation, user questionnaires, and tests to assess learning. PSE support for creativity can be evaluated through the use of these self-report questionnaires and through the analysis of the PSE's support for the workflow of problem solving. A PSE evaluation should be considered a success if the PSE supports creativity and meets or exceeds the requirements of formal usability testing of the system. As any application used by a user may influences the end product of their work, creativity support should be a required component for the successful usability evaluation of any system, not only problem solving environments.

When evaluating a PSE, support for creativity must be measured with respect to the support provided by the original environment. Ethnographic studies should be conducted to classify the methods and information forms used in problem solving and to identify where computers have and have not been used in users' work.

Using the workflows present in Figures 1 and 2, one can classify how the PSE supports problem solving and flow. This involves specifying for each conversion and modification what support the PSE provides the user. For instance, if Jim's simulator is termed a PSE, it only provides support for executing a simulation, a partial support of c10, result formation. Such a classification scheme allows for clear definitions of what a PSE is and assists in the process of PSE development. This assistance is through providing a model, which developers can refer to when performing ethnographic studies of users. It gives points to look for and a method of describing workflow in a group. Furthermore, it provides a framework for the use of scenario-based design [2] by suggesting possible scenarios for PSE use. The scenario involving Jim suggests several scenarios for his usage of a PSE.

RELATION TO GENEX FRAMEWORK

GENEX (Generator of Excellence) [26] is a four-phased integrated framework for the support of creativity. These four phases of user work in GENEX (Collect, Create, Relate, Donate) are supported by WorkFlow through the applications associated with the conversion and modifier processes. WorkFlow differs from GENEX in that it seeks to further clarify the work involved in the creative process, and to associate the information forms used in creative activities with specific places in the overall workflow of problem solving. This is a useful distinction because the same set of activities or strategies for problem solving are not present among all problem solvers. The application of a strategy for supporting individuals in a PSE should not be based on generalized software support for the activities of a certain class of problem solvers. This paper also provides a method for the evaluation of a PSE's support for creativity. In essence, our theory treats creativity at a lower level than GENEX, since it seeks to provide a basis for the construction of PSEs that support creativity. WorkFlow also seeks to support creativity beyond evolutionary creativity by supporting creativity in which new models are built to solve problems.

CONCLUSION AND FUTURE WORK

Whether WorkFlow leads to creative accomplishments by PSE users will not be known for some time. However, the development of a theory of the creative problem solving user enables more articulate work in the design and evaluation of PSEs, thereby advancing PSE usability.

Further research is needed to refine WorkFlow theory and to provide a more concrete methodology for the evaluation of creativity in PSEs. Empirical studies of WorkFlow's effectiveness in supporting creativity can then be performed using existing PSEs. WorkFlow is currently being applied and refined in the development of JigCell, a PSE for modeling the eukaryotic cell cycle.

It is our goal to continue to apply WorkFlow and to assist both users and developers in using and creating the best environments for creative problem solving.

ACKNOWLEDGEMENTS

We would like to thank the PSE group at Virginia Tech and the students of CS 5724 and CS 6724 for their many helpful comments and suggestions.

REFERENCES

1. Campbell, D., Blind variation and selective retention in creative thought as in other knowledge processes. Psychological Review, 1960(67): p. 380-400.
2. Carroll, J.M., *Making use: Scenario-based design of human-computer interactions*. 2000, Cambridge: MIT Press.
3. Csikszentmihalyi, M., *Beyond boredom and anxiety*. 1977, San Francisco: Jossey-Bass.
4. Csikszentmihalyi, M., *Creativity: Flow and the psychology of discovery and invention*. 1996, New York: HarperCollins.
5. Csikszentmihalyi, M., *Flow: The psychology of optimal experience*. 1990, New York: Harper & Row.
6. Csikszentmihalyi, M., *Implications of a systems perspective for the study of creativity*, in *Handbook of Creativity*, R.J. Sternberg, Editor. 1999, Cambridge University Press: New York. p. 313-335.
7. Csikszentmihalyi, M. and J. LeFevre, *Optimal experience in work and leisure.* Journal of Personality and Social Psychology, 1989. **56**(5): p. 815-822.

8. Dictionaries, A.H., ed. *The American Heritage® Dictionary of the English Language*. 4 ed. 2000, Houghton Mifflin: Boston.
9. Ellis, G.D., J.E. Voelkl, and C. Morris, *Measurement and analysis issues with explanation of variance in daily experience using the flow model.* Journal of Leisure Research, 1994. **26**(4): p. 337-356.
10. Finke, R.A., T.B. Ward, and S.M. Smith, *Creative cognition: Theory, research, and applications*. 1992, Cambridge: MIT Press.
11. Fishwick, P.A., *A taxonomy for simulation modeling based on programming language principles.* IIE Transactions, 1998. **30**: p. 811-820.
12. Gallopoulos, E., E. Houstis, and J.R. Rice, *Computer as Thinker/Doer: Problem-Solving Environments for Computational Science.* IEEE Computational Science & Engineering, 1994. **1**(2): p. 11-23.
13. Getzels, J.W. and M. Csikszentmihalyi, *From problem solving to problem finding*, in *Perspectives in creativity*, J.W. Getzels, Editor. 1975, Aldine: Chicago. p. 90-116.
14. Ghani, J.A. and S.P. Deshpande, *Task characteristics and the experience of optimal flow in Human-Computer Interaction.* The Journal of Psychology, 1994. **128**(4): p. 381-391.
15. Ghani, J.A., R. Supnick, and P. Rooney. *The experience of flow in computer-mediated and in face-to-face groups*. in *Twelfth International Conference on Information Systems*. 1991. New York, NY.
16. Hoffman, D.L. and T.P. Novak, *Marketing in computer-mediated environments: Conceptual foundations.* Journal of Marketing, 1996. **60**(July): p. 50-68.
17. Holland, J.H., et al., *Induction: Processes of inference, learning, and discovery*. 1986, Cambridge: MIT Press.
18. Katz, J.A., *Playing at Innovation in the computer revolution*, in *Psychological issues of human computer interaction in the work place*, M. Frese, E. Ulich, and W. Dzida, Editors. 1987, North-Holland: New York.
19. Maslow, A.H., *The farther reaches of human nature*. 1971, New York: Viking.
20. Nakamura, J. and M. Csikszentmihalyi, *Catalytic creativity: The case of Linus Pauling.* American Psychologist, 2001. **56**(4): p. 360-362.
21. Nickerson, R.S., *Enhancing Creativity*, in *Handbook of creativity*, R.J. Sternberg, Editor. 1999, Cambridge University Press: New York. p. 392-430.
22. Novak, T.P., D.L. Hoffman, and Y.F. Yung, *Measuring the customer experience in online environments: A structural modeling approach.* Marketing Science, 2000. **19**(1): p. 22-42.
23. Rice, J.R. and R.F. Boisvert, *From scientific software libraries to problem-solving environments.* IEEE Computational Science & Engineering, 1996. **3**(3): p. 44-53.
24. Rogers, C.R., *On becoming a person*. 1961, Boston: Houghton Mifflin.
25. Runco, M.A., *Conclusions concerning problem finding, problem solving, and creativity*, in *Problem finding, problem solving, and creativity*, M.A. Runco, Editor. 1994, Ablex: Norwood. p. 272-290.
26. Shneiderman, B., *Creating creativity: User interfaces for supporting innovation.* ACM Transactions on Computer-Human Interaction, 2000. **7**(1): p. 114-138.
27. Tyson, J.J., K. Chen, and B. Novak, *Network Dynamics and Cell Physiology.* Nature Reviews Molecular Cell Biology, 2001(2): p. 908-916.
28. Vass, M., et al., *The JigCell Model Builder*, http://csgrad.cs.vt.edu/~mvass/pse/ModelBuilder.pdf, 2002.
29. Vass, M. and P. Schoenhoff, *Error detection support in a cellular modeling EUP environment*, To appear in *Proceedings of the 2002 IEEE Symposium on Human-Centric Languages and Environments*, 2002.
30. Ward, T.B., S.M. Smith, and R.A. Finke, *Creative cognition*, in *Handbook of creativity*, R.J. Sternberg, Editor. 1999, Cambridge University Press: New York. p. 189-212.
31. Webster, J., L.K. Trevino, and L. Ryan, *The dimensionality and correlates of flow in human computer interactions.* Computers in Human Behavior, 1993. **9**(4): p. 411-426.