1-1-1993

Visual Process Language: An Environment That Enhances Inventiveness in the System Development Process -- a Theoretical Perspective

Wita Wojtkowski
Boise State University

Waclaw G. Wojtkowski
Boise State University

VISUAL PROCESS LANGUAGE: AN ENVIRONMENT THAT ENHANCES INVENTIVENESS IN THE SYSTEM DEVELOPMENT PROCESS-- A THEORETICAL PERSPECTIVE

Wita Wojtkowski & W. Gregory Wojtkowski
College of Business
Boise State University
1910 University Drive
Boise, ID 83725
(208) 385-1372; BITNET: RISWOJT2@IDBSU

ABSTRACT

This paper provides a theoretical perspective on the effect of computer support on creativity. The assertion is made that artifacts of IS do not substantially differ from those of art and science, and that the creative processes leading to the artifacts are similar. It further proposes that the use of newly available computer technology, specifically, the Visual Process Language (VPL) will have much to offer in terms of augmenting creativity when constructing IS artifacts. We postulate that VPL assists the first insight, preparation and illumination steps of the creative process, and that both Association/Images and Progressive Abstraction techniques are embedded in VPL.

1. Introduction

> *All creative activity is exploration, gambling, and adventure. The important thing is not so much to obtain new facts as to discover a new ways of thinking about them.*
> *Attributed to an artist or a scientist*

Research on creativity has cumulatively pointed out the importance of inventiveness enhancement techniques in creating artifacts. Whereas the nominal methods have received much attention, studies of the effect of computer support on creativity are scarce [Elam et al., 1990]. There are many possible explanations that could account for this. One of the probable reasons is a lack of theory in guiding such studies. It is the intent of this paper to fill in this gap.

This paper proposes that the use of newly available computer technology, Visual Process Language (VPL) will have much to offer in terms of augmenting creativity when developing computer information systems. We assert that the system development environment embodied in VPL explicitly incorporates creativity enhancing procedures, specifically, the Association/Images and Progressive Abstraction techniques.

Moreover, VPL offers a innovative platform for assessment of creativity improvement in the development of information systems. Although at present VPL is used only in a form of a wholly functional prototype [Shepard et al., 1992], we propose that its appraisal, from the point of view of process creativity enhancement, can substantially assist our understanding of the mechanisms of inventiveness in IS.

2. The Proposition

Creativity is manifested in art as well as in science. The manifestations of an individual (or team's) creativity are to be found in the artifacts produced. However, an illogical proposition often slips within comparative discussions of creativity in art and science.

This proposition, may be stated as follows. The art form X will not exist without the artist Y, whereas the theory A would exist without a scientist B; these two percepts therefore reveal a fundamental difference between art and science.
The following statement serves as concrete example of the precept of the proposition:

> *AN AMERICAN IN PARIS will not exist without George Gershwin, whereas the Theory of Object-Orientation will exist without Bjarne Stroustrup*

It is possible to generate an indefinite number of such statements simply by inserting appropriate terms for X, Y, A, and B. The proposition is erroneous because it inappropriately parallels an individual artifact--an outcome of the

creative process, with the theory--an outcome of distillations and extrapolations of the collections of artifacts. This parallel fallacy [Topper, 1981] is made between unequal factors. A correct parallel would be artifact-to-artifact rather than artifact-to-theory. This is summarized in Figure 1.

The first parallel involves the creators (makers and doers) themselves.

The second parallel consists of what may be called the objects of the world being represented or modeled by the creators (factors of the domain). [1]

The next parallel is crucial to the argument. Here is the locus of the fallacy in the logic of the initial proposition.

The reason is this: the oil painting *Mona Lisa* is the artistic artifact. As an individual artifact it would not exist without Leonardo da Vinci. It is an oil painting of specific size, choices of colors, composition, and choice of background. In the proposition (and in its diverse expressions such as examples a., b., and c.) the artifact is paralleled with a theory [2], which is an extraction and extrapolation from a collection of artifacts. The artifact parallels for Da Vinci's painting are Einstein's pivotal paper, Stroustrup's C++ language, and team developed VPL. [3]

3. An Overview of Creativity Literature in IS

The literature on creativity and the creative process is quite extensive. A relatively large amount of writing on the subject (especially concerning the model of the creative process) occurred in 1960 and 1970. The summary in Figure 2 list references for the IS domain. The references are grouped according to their principal concern with:
- the nature of the process (descriptive approach: an attempt to understand what goes on)
- nurture of the process (prescriptive approach: describing what should be done to create effectively)
- the creative artifact

3.1 The theoretical model
Model of creativity process A simplified model of the creativity process is shown in Figure 3. This model applies regardless of the domain (Wallas, 1926; Ghiselin, 1963; Kneller, 1965; Lawson, 1980).

Hypothesis 1:

The core of scientific and art creativity is to be found in their artifacts. The creative processes leading to the artifacts are similar.

3.2 Creativity in IS
The information systems development process can be depicted as an effort aimed at creation of artifacts specific to the IS domain. A developmental process entails artifacts specific to it: certain set of files, state data, and so on. At any given time a subset of these artifacts is worked on by those who are involved in system development. Creativity in developing information systems is conceptualized as a means through which one, two, or more purposive actors arrive at an innovative artifact. Actors are all those involved in the system development. Depending on the development model [4], these can be system analysts and developers as well as end-users. The means are constrained by the specific development environment. In the case of more than one actor, the process is many-sided. The delegation of work to individuals, and the tracking of the responsibilities involved is an important aspect of the process (for example in Joint Applications Development (JAD) [Wood and Silver, 1989; when Computer Aided Software Engineering (CASE) is utilized in a LAN environment [Tate et al. 1992]).

3.2.1 First insight
First insight involves the recognition that the problem or opportunity exists and a commitment is made to solving it. In the area of human activity systems, which underpin the development of information systems, it is not always possible to describe initial conditions fully.

3.2.2 Preparation
Preparation involves a deliberate effort to develop an idea for solving the problem. Various information systems analysis and design methods are mechanisms enabling this "deliberate effort."

3.2.3 Incubation
This step of the process, in most writings on creativity, is considered to involve no apparent effort. It is possible to postulate, however, that some of the new technologies which can be used in the preparation step aid incubation as well.

3.2.4 Illumination
This step represents a sudden emergence of an idea through reorganization and reexamination of the preparatory stage. The development parameters

360

Domain	Art	Hard science	IS: individual effort	IS:team effort
Originator	Leonardo da Vinci	Albert Einstein	Bjarne Stroustrup	Shepard et al.
Factors of the domain	Mona Lisa	Moving bodies	Data in object sense	Software development processes
An individual artefact	Oil painting of Mona Lisa	1905 paper in *The Annalen der Physics* "On the Electrodynamics of Moving Bodies"	C++	VPL A Visual Process Language
Theory	Treatises on painting and perspective	Theory of relativity	Theory of object orientation	Theory of systems development processes

FIGURE 1 APPROPRIATE PARALLELS FOR ART AND SCIENCE

| Domain	NATURE of creativity	NURTURE of creativity	The artifact
IS related	Argyris, 1977 Argyris et al., 1978 Maidique, 1980 Volcema, 1986 Cowen, 1986 Langley et al., 1987 Vail, 1989 Couger, 1989 a) Couger, 1990 b) Couger et al., 1990 a)&b) Galetta et al., 1991 Duncan et al., 1992	Quinn, 1985 Nunamaker et al., 1987 Salaway, 1987 Freedman, 1987 Couger, 1989 b) Cobbin et al., 1989 Couger, 1990 a)&c) Van Gundy, 1991 Sampler et al., 1991 Snow et al., 1991 Couger, 1991 Costello, 1991 Galetta et al., 1992	Jackson, 1965 Besemer et al., 1981 Rijesenbij, 1990 Couger et al., 1991 Evans, 1991 Couger et al., 1992

FIGURE 2 LITERATURE CITATIONS FOR CREATIVITY IN IS
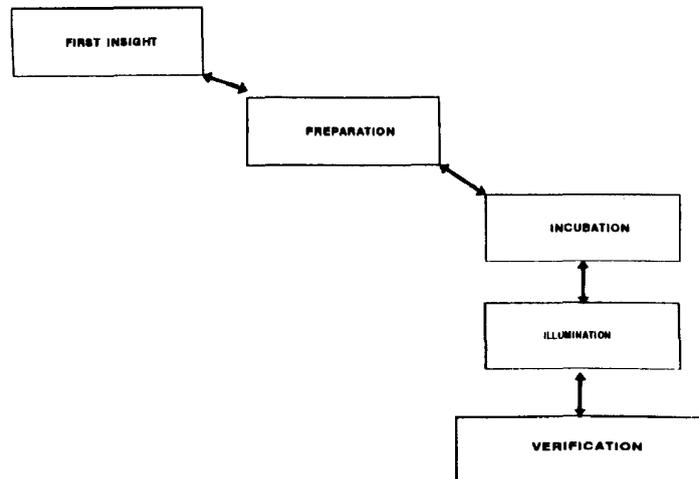
FIGURE 3 MODEL OF THE CREATIVE PROCESS

361

that unfold throughout the process may aid this step.

3.2.5 Verification

This step involves full system development and testing. Many automated tools (fourth generation languages and CASE, for example) aid this step.

3.3 IS artifacts

The focus of a creative process is creation of the artifact. At first approximation, it is possible to assert that the artifacts of art are concerned with form, whereas artifacts of science are focused on the content. But a moment's reflection reveals that this content/form distinction is problematic. The artifacts of any significance, regardless of the domain, are often acknowledged for the content as well as the form.[5] In addition, the artifacts of IS (and many of the artifacts of applied sciences) have to exhibit utility and be perceived by others to be not only novel, but also useful [Jackson and Messick, 1965; Besemer, 1981; Evans, 1991, Couger et al., 1992][6].

4. Creativity Support System

Couger and Dengate provide examples of different levels of creative opportunities for various tasks provided in IS organization (Couger and Dengate, 1992, p. 290). Understandably, system development is listed as offering a high degree of creative opportunity since it embodies the process of creating IS artifacts. Any IS development effort includes a set of files, state data, and other artifacts, a complete set or subset of which is eventually handled by users.

Approaches to creativity in general, and within IS in particular, are either descriptive or prescriptive in nature. The descriptive focusing on the nature of the process, attempts to provide an understanding of what goes on during the process of creation. The prescriptive focusing on the nurture, indicates what should be done to create 'better' artifacts. This paper proposes that the use of newly available computer technology (Visual Process Language, VPL) will have much to offer in terms of augmenting creativity when constructing IS artifacts.

4.1 System development process model

There exists a great diversity of possible models of the information systems development process. (In practice, every system developer, or development organization uses a somewhat different

model [Agresti, 1986; Armenise, 1989].) These diverse process models include the prototyping model [Maude and Willis, 1991], the 'fountain' object-oriented model [Henderson-Sellers et al., 1990], the spiral model [Boehm, 1988], the transformational and the operational specification models [Agresti, 1986], and many others [Mayer, 1989].

At the same time, the practice of systems development is becoming more complex and capital intensive. For example, LAN based team development is rapidly replacing other configurations. Today, the major concern of new development 'ideologies' is the problem of scaling up for larger and more complex projects involving many developers [Forte and Norman, 1992].

4.2 A VPL

In this situation, the ability to create a completely customizable model of the system development process becomes essential [Osterweil, 1987]. Information from such an adaptable process model can then be used to:

.   automate some portions of the process
.   guide other portions

Formal depiction of a customizable development process that allows computer support for enaction is now attainable in a Visual Process Language (VPL). VPL combines a conceptually intuitive visual process modelling language with a process programming language [Shepard et al., 1992].

A process modeling language is based on the set of rules which determine the process. A process programming language is a formal enactable model of the process. The process model is said to be enacted [Sheppard et al., 1992, p. 38] " . . . when some mechanism uses a supplied process program to monitor the progress of the many concurrent streams of a development effort. It proposes the invocation of tools at appropriate times, enforces the process model, gives guidance to the users . . . and executes the completely automated activities in the process model."

A system development process model ought to embody activities that:

.   are performed completely by humans, like the creation of a modular decomposition
.   are performed completely by the computer, for example, compiling
.   require both human work and support of the computerized tools, for example text editing

362

Although there may be some aspects of the process that are difficult or even impossible to formalize, formal description of the process is required to allow computer support for enaction. Thus the information from the enacted process model serves to automate some portions of the process and guides the others. The guidance aspect of the VPL, a language for both process modeling and process programming, allows its users ". . . the flexibility to develop creatively while controlling process shortcuts. . . [Sheppard et al., 1992, p. 43].

4.2.1 Functionalities

VPL essentially does two things. It 1) visually represents; and 2) enables the enaction of the development processes. The visual representation and enaction are accomplished in VPL through a combination of certain features of the logic flowcharts, petri nets, and object paradigms. The development process is depicted as an effort aimed at a creation of artifacts. A given process comprises artifacts specific to it: sets of files, state data, and so on. At any given time a subset of these artifacts is worked on by the systems developers.

An object in a VPL program consist of all the artifacts associated with a currently active developmental work task. A part of the developmental work may be canceled by deactivating all objects associated with it. Currently active objects are stored in the object table. The results of one work task can be used as a starting point for the next one.

In a VPL program there are four mechanisms acting on objects:

. creation . advancement
. reactivation . deactivation

Enaction of a VPL program is accomplished through these mechanisms. Advancement is a flow of objects through a graph of nodes which represents visual depiction of the process.
The nodes are of nine types:

. start . branch
. task . decompose
. procedure . recompose
. split . finish

Graphical representation of VPL node types is shown in Figure 4.

A program start node represents the point at which objects are created. This node is also the entry point for objects from the outside of the process. A task node represents an action performed on the object by an automated tool only, a user[7], a tool and a user. This node encapsulates information about a
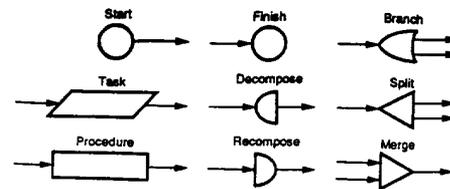
Figure 4 GRAPHICAL REPRESENTATION
OF THE VPL NODES

an automated tool that is part of the development process. A task node may also embody the location of the interface point between an off-line activity and the development environment. Organizational meetings of those involved in a specific stage of the process may serve as an example. An electronic record of the minutes of such a meeting, and the decision reached, are then entered into a VPL program.

A procedure node represents a group of nodes that have a common conceptual objective. Thus, the top level of each process program is depicted as a root procedure.

A split node creates full duplicates of the object entering this node. On the output from the node, each copy follows a different process. For example, program module specification may enter this node. On the output, copies of this module will follow different procedures. Eventually, various versions of the program might be created.

A merge node is a collection point for the concurrent flow of objects. A typical use might be to select the best object and deactivate the others.   A branch node allows the object to emerge along only one of the available outputs. Decision as to which output an object follows might depend, for example, on user-prompted replies.

Decompose and recompose nodes deal with a family of objects. An object entering the decompose node (parent) emerges as a family of objects. Each of the objects in this family will have only part of the parent's object information. An example of use for this node might be the assignment of parts of a large program to individual programmers. Objects entering the recompose node accumulate at the input until the family is complete. At the output they emerge as a single, larger object. A typical use might be to initiate large program testing.

Finish node signifies completion of the process program or of the procedure. Objects that enter the process program finish node are deactivated. A project or its part may be canceled by deactivating all objects associated with it.

363

The information associated with an object changes from node to node, thus the type of an object in VPL is determined by the node at which it currently resides. The nodes's action may change the object's state, or even number of objects. Moreover, an object's deactivation and reactivation timing are not know a priori. For this reason, these two mechanisms are built into an enactor which is separate from the VPL program, and must be invoked by the user.

All individuals on a development project are assigned certain roles. These roles are part of the process model. They include end-users, analysts, designers, programmers, and so on. At a given time individuals have exactly one role. Every time they enter the VPL environment they must select which role they want to assume. The role they assume at a given time determines the inventory of actions available to them. These are stored in a role table and constitute the user model.

Specific work assignments enter the process program as the new objects assigned to roles. An active object in a VPL is precisely one work assignment for a person acting in a specific role. The object exists as a result of object creation or reactivation. A VPL environment is shown in Figure 5.

4.3 VPL Relevance to Creativity Enhancement

In this combination of visual process modeling and process programming resides the potential for support and enhancement of creativity in the systems development process. At present VPL is implemented and evaluated in the form of a fully functional prototype. Its strengths are these [Shepard et al., 1992]:

. it places few constrains on the tools used in the development process
. it places few constrains on the process itself
. it allows users the flexibility to develop creatively while controlling process shortcuts
. it demonstrates the plausible alliance of process modeling and process programming

4.3.1 How VPL assists creativity in IS

Two IS creativity enhancement techniques, namely Association/Images and the Progressive Abstraction [Couger et al., 1991] are relevant in the context of VPL. Association/Images technique uses associative and visual thinking as well as linking and

coupling of processes to expand solution space. Progressive Abstraction technique generates alternative problem definition gradually transferring to higher levels of problem abstraction through links and encapsulation of problem substructures.

We assert that both techniques are embedded in VPL and arrive at the following:

Hypothesis 2:

VPL enhances creativity in the development of IS artifacts through Association/Images and the Progressive Abstraction techniques.

To illustrate this position, we first propose to consider VPL as an automated decision and inventiveness aid. The VPL environment is composed of the following: process model, tool model, user interface, and data model. The types of support and required functionalities [Zachary, 1986] to be provided by the automated aids are listed in Figure 6.

VPL conforms to all required functionalities. In addition, VPL might simulate the development process model of any developer or maintainer, with few constraints on the tools or the process itself. This is quite an achievement, since there exists such a vast diversity of possible models of systems development. With VPL, the development process model can combine activities:

1. Performed completely by the computer (for example, compiling)
2. Performed completely by humans (for example, creation of modular decomposition)
3. A combination of human expert decisions aided by computerized tools (for example, text editing)

Information control techniques and representation aids embedded in the VPL allow the software process model, when enacted, to monitor the progress of many concurrent sequences of the development effort. In the VPL environment it is possible to automatically obtain counsel on tools and their use at the at appropriate stage of the process. The process model itself is automatically enforced and guidance furnished. The information on the status of the various developmental sequences is automatically available as well.

For example, let's consider the information system change model as represented in VPL (Figure 7).

When the system change is initiated, at the start node the new object created using a specialized tool that is part of this environment. This new object

364
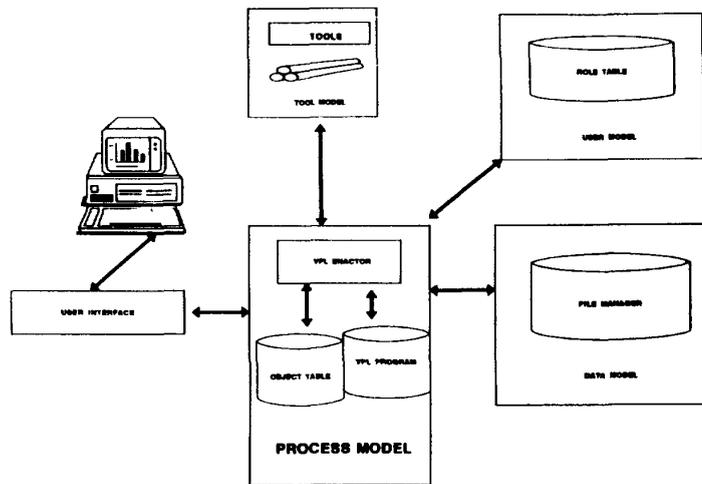
FIGURE 5 VPL ENVIRONMENT MODEL

Assistance with	Purpose
process models	representation and approximation of complex processes
choice models	integration of individual criteria across varied features or alternative choices
information control techniques	storage, retrieval, and organization of information and knowledge needed for a resolution
representation aids	presentation and manipulation of a specific representation of the decision problem
analysis and reasoning aids	execution of problem specific reasoning processes based on the representation of a problem

FIGURE 6 TYPE OF SUPPORT AND REQUIRED FUNCTIONALITIES
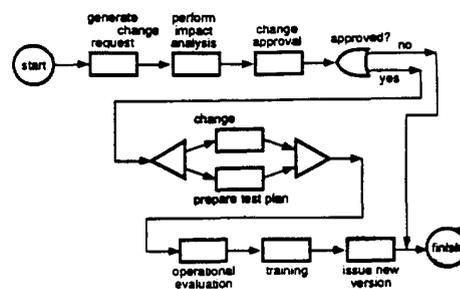
FIGURE 7 SYSTEM CHANGE MODEL AS REPRESENTED IN THE VPL

365

might consist of a rough statement of work required. When the object has been created, it advances automatically to the next node. In the generate change request node, creative human intervention is required. Specifically, this node might contain separate subprocedures which compel the user to specify new capabilities and define documentation changes. The events comprising node enaction rely on user intervention in such a way that they allow flexibility to develop creatively while supporting all the phases of the applicable development process.

We also assert that VPL assists the first insight, preparation and illumination steps of the creative process. Specifically:

. Although a formal description of the processes is needed to allow automated process support, VPL conventions recognize that there are some aspects of some processes that are impossible to formalize. This insight forces the users of VPL to find inventive ways of solving the difficulty.

. VPL allows for diverse approaches to the same problem to be enacted simultaneously. Exercising different process options may trigger the sudden emergence of an idea on how to eventually proceed. This is especially useful for complex development effort, and facilitates both preparation and illumination.

Development of complex systems involves a multitude of substantial information processing activities [Armenise, 1989]. As discussed, VPL through its functionalities might substantially aid the inventiveness in the systems development process. We envision this situation as in a schematic shown in Figure 8.

5.   Concluding Remarks

We assert that the artifacts of IS do not substantially differ from those of art and science ("hard" as well as applied). Creativity and creation of the artifacts in art and science are richly researched topics. But only recently has the IS literature focused on the creative problem solving and efficacy of creativity techniques (see comments in Galetta et al., 1992, Couger et al., 1992). The systems development environment embodied in the VPL explicitly incorporates creativity enhancing procedures. Specifically, these are, the Association/Images and

Progressive Abstraction techniques. VPL also might offer a platform for evaluation and assessment of creativity improvement in the development of information systems.

Although at present VPL is used only in a form of a fully functional prototype Shepard et al., 1992], its appraisal from the point of view of process creativity enhancement can substantially assist our understanding of the mechanisms of inventiveness in IS development. Further research should focus on how one might perform an empirical evaluation of the improvement of the development processes. This will require establishment of the metrics construct that measure inventiveness of the design, for example. Perhaps with the VPL environment, it will be possible to asses such metrics - we envision the relationships (Figure 9) as a modification of those shown in Figure 8. This structure (adapted from Tate et al., 1992) proposes a formal framework for understanding the creative improvement of the development processes.

6.   References

Agresti, W., New Paradigms for Software Development, IEEE Computer Society Press, 1986.

Argyris C., Schon, D., Organizational Learning: A Theory of Action Perspective, Addison Wesley, Reading, 1978.

Argyris, C., "Double loop learning in organizations," Harvard Business Review, September/October, 1977, pp. 115-125.

Armenise, P. "Process software machines: a framework for future software development environments," Proceedings of the Second European Software Engineering Conference, September 1989, pp. 118-139.

Besemer, S. P., Treffinger, D., J., "Analysis of Creative Products: Review and Synthesis," The Journal of Creative Behavior, November 15, 1981, pp. 158-178.

Boehm, B. W., "A spiral model of software development and enhancement," IEEE Computing, May 1988, pp. 61-71.

Cobbin, W. F., Kozar, K. A., Michaele, S. J., "Establishing telemarketing leadership through information management: Creative concepts at AT & T American Transtech," MIS Quarterly, Vol. 13, September 1989, pp. 360-372.

Costello, J., "Radical ways to help projects: How does a software developer avoid boredom and speed up project times?", Computer Weekly, June 20, 1991, pp. 30-31.

Couger, J. D., "New challenges in motivating MIS personnel," Journal of Information Systems Management, Vol. 6, Fall 1989 a), pp. 36-41.

Couger, J. D., "Ensuring creative approaches in information system design," Center for Research on Creativity Report 89-2, University of Colorado, Colorado Springs, 1989 b).

Couger, J. D., "Motivating analysts and programmers," Computerworld, Vol. 24, January 15, 1990 a), pp. 73-76.

Couger, J. D., "Creativity in IS," Computerworld, Vol. 24, October 29, 1990 b), pp. 123-124.

Couger, J. D., "Ensuring creative approaches in information systems design," Managerial and Decision Economics, Vol. 11, 1990 c), pp. 281-295.

Couger, J. D., Adelsberger, H., Borovits, I., Zviran, M., Motiwalla, J., "Commonalities in motivating environments for programmer/analysts in Australia, Israel, Singapore, and the USA," Information and Management (Netherlands), Vol. 18, January 1990 a), pp. 41-46.

Couger, J. D., Higgins, L. F., McIntyre, S. C., "Differentiating Creativity, innovation, entrepreneurship, intrapreneurship, copy right and patenting for IS product/processes," Proceedings of the 23th Hawaii International Conference on System Sciences, 1990 b), pp. 281-285.

Couger, J. D., "Using bottom-up approach to creativity improvement in IS development," Journal of Systems Management, Vol. 42, September 1991, pp. 23-27.

Couger, J. D., Higgins, L. F., McIntyre, S. C., "Preliminary study: Measurement of creativity in IS products," Report CR91-6, Center For Creativity and Innovation, Colorado Springs, 1991.

Couger, J. D., Dengate, G., "Measurement of creativity of IS products," Proceedings of the 25th Hawaii International Conference on System Sciences, 1992, pp. 288-296.

Cowan, D., "Developing process model of problem recognition," Academy of Management Review, Vol. 11, 1986, pp. 763-776.

Duncan, N., Paradice, D., "Creativity in GDSS: An integration of creativity, group dynamics, and problem solving theory," Proceedings of the 25th Hawaii International Conference on System Sciences, 1992, pp. 277-287.

Elam, J., J., Mead, M., "Can Software Influence Creativity?" Information Systems Research, Vol. 1, March 1990, pp. 1-22.

Evans, J., R., Creative Thinking in the Decision and Management Sciences, South-Western Publishing, Cincinnati, 1991.

Forte G., Norman, R. J., "A self-assessment by the software engineering community," Communications of the ACM, April 1992, pp. 28-32.

Freedman, D. H., "Cultivating IS creativity," Infosystems, Vol. 34, July 1987, pp. 24-27.

Galetta, D., F., Sampler J. L., Teng, J. T. C., "Strategies for integrating creativity principles into the system development process," Proceedings of the 25th Hawaii International Conference on System Sciences, 1992, pp. 268-275.

Galetta, D. F., Sampler, J. L., "Creativity: The missing ingredient in identifying strategic information systems," Working Paper 703RR, University of Pittsburgh, Graduate School of Business, 1991.

Ghiselin, B., (ed.) The Creative Process, Mentor, New York, 1963.

Gogan, J., L., Cash, J. I., "IT based Innovation," Proceedings of the 25th Hawaii International Conference on System Sciences, 1992, pp. 257-267.

Henderson-Sellers, B., Edwards, J. M., "The object oriented life cycle," Communications of the ACM, September 1990, pp. 142-159.

Jackson, P., Messick, S., "The person, the product and the response: conceptual problems in the assessment of creativity," Journal of Personality, Vol. 33, 1965, pp. 309-329.

Kneller, G. F., The Art and Science of Creativity, Holt, Rinehart, and Winston, New York, 1965.

Langley, P., Simon, H., A., Bradshaw, G. L., Zytkow, J. M., Scientific Discovery: Computational Explorations of the Creative Processes, MIT Press, Cambridge, 1987.

Lawson, B., Creative Thinking, Architectural Press, Guilford, 1990.

Maidique, M., " Entrepreneurs, champions, and technological innovation," Sloan Management Review, Winter 1980, pp. 59-75.

Markus, A., Graphic Design for Electronic Documents and User Interfaces, ACM Press, 1992.

Maude, T., Willis, G., Rapid Prototyping, Pitman, London, 1991.

McKim, R., Experiences in Visual Thinking, Brooks & Cole, Monterey, 1972.

Meyer, B., "From structured programming to object-oriented design: The road to Eiffel," Structured Programming, Vol. 1, 1989, pp. 19-39.

Miller, A. I., Albert Einstein's Special Theory of Relativity, Addison-Wesley, 1981.

Nunamaker, J. F., Applegate, L. M., Konsynski, B. R., "Facilitating group creativity: Experience with group decision support system," Journal of Management Information Systems, Vol. 3, Spring 1987, pp. 5-19.

Osterweil, L., "Software processes are software too," Proceedings of the Ninth International Conference on Software Engineering, March 1987, pp. 2-13.

Qiunn, J., B., "Managing innovation: Controlled chaos," Harvard Business Review, May/June, 1985, pp. 73-84.

Rijsenbrij, D., "Foundations of quality," Systems International (UK), Vol. 18, January 1990, pp. 22-24.

Salaway, G., "An organizational learning approach to information systems development," MIS Quarterly, June 1987, pp. 245-264.

Sampler, J.L., Galetta, D. F., "Individual and organizational changes necessary for the application of creativity techniques in the development of information systems," Proceedings of the 24th Hawaii International Conference on System Sciences, 1991, pp. 404-411,

Shepard, T., Sibbald, S., Wortley, C., "A visual software process language," Communications of the ACM, Vol. 35, April 1992, pp. 37-43.

Snow, T. A., Couger, J. D., " Creativity improvement intervention in a system development work unit," Proceedings of the 24th Hawaii International Conference on System Sciences, 1991, pp. 412-418.

Tate, G., Verner, J., Jeffrey R., "CASE" A testbead for modeling, measurement and management," Communications of the ACM, April 1992, pp. 65-72.

Topper, D. "On the creativity and discovery in the fine arts and in the natural sciences," Leonardo, Vol. 14, No. 2, 1981, pp. 144-146.

Vail, P., B., Managing as a Performing Art, Jossey Bass, San Francisco, 1989.

Van Gundy, A. B., Techniques of Structured Problem Solving, Van Norstrand, New York, 1991.

Wallas, G., The Art of Thought, Harcourt, Brace, New York, 1926.

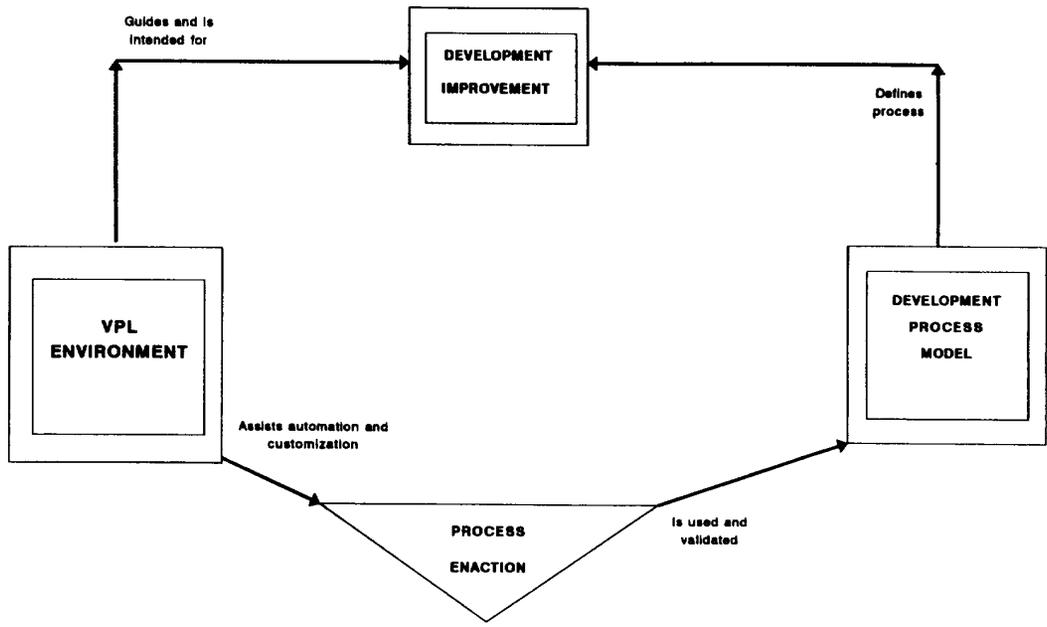Wood J., Silver, D., Joint Application Design, John Wiley, New York, 1989.

367

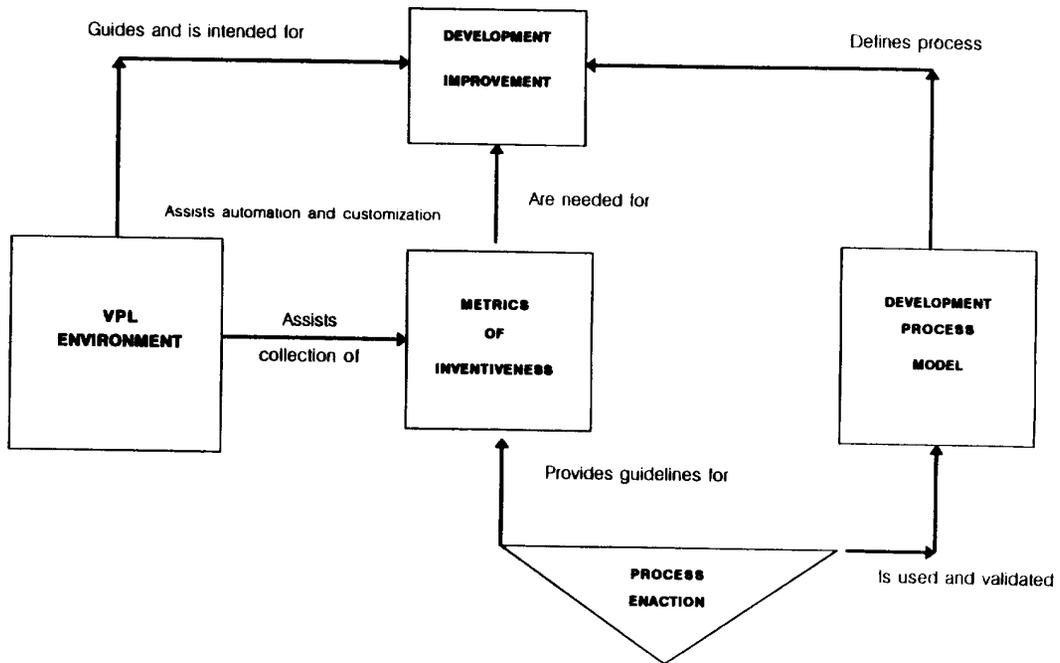**FIGURE 8 VPL ENVIRONMENTAL AND THE DEVELOPMENT PROCESS**

FIGURE 9 VPL ENVIRONMENT AND IMPROVEMENT OF THE DEVELOPMENT PROCESS

368