BEHAVIORAL SCIENCE RELEVANCE TO SOFTWARE SYSTEMS ENGINEERING AND DESIGN

It is unlikely that any major benefit to software systems engineering and design can be discovered, unless researchers become familiar with the relevance of selected theory and empirical evidence coming from or closely allied to behavioral science.

There is both theoretical and empirical evidence to support the foregoing statement.

Empirical Evidence. Three distinct pieces of empirical evidence support the foregoing:

A. Superprogrammer Behavior. Studies carried out in West Germany have shown the existence of "superprogrammers" who develop 15 to 30 times as much computer code per day as average programmers, and whose code is higher quality than that of the average programmer.

Personality profiles of these individuals are distinctly different from those of the average programmer. These individuals show behavior very similar to that identified in the literature of "creativity"; namely they do their best work in environments not commonly encountered in bureaucratic business organizations. They will not work for large companies, but instead are independent workers.

B. Chief Programmer Teams. Chief Programmer Teams were introduced around 1972. Such teams have been shown to have the capability for "significantly improving productivity". The organization of such a team follows good behavioral practice. Each role that is needed in development of software is distinguished, and members of the team take roles for which they are particularly qualified. The Chief Programmer coordinates team activity, and may also bring in concepts from the "egoless team" idea.

C. Egoless Teams. The concept of "egoless team" originally was to take advantage of structured programming as a way to partition tasks so that the work of programming was shared by a team of equals. This concept did not work very well because of inadequate leadership. More recently the concept has been expanded whereby the egoless team follows a process that helps assure high quality products. With this process, work is structured as in the Chief Programmer Team, but the work of each individual is scrutinized in small modules by each other member of the team. Barry Clemson has reported that 15,000 lines
of code produced in this way was found to contain only two small "bugs", which is a vast improvement over individualistic programming activity. Also the documentation is inherently promoted by this system, since the programmer is required to make his work comprehensible in small quantities to other members of the team.

Theoretical Evidence. Extensive conflict in all kinds of organizations, and especially among nations who possess nuclear weaponry sufficient to destroy the earth has led systems scientists and cyberneticians to delve into why people are unable to reach agreement. There is no comprehensive theory as yet, but part of the product of this kind of investigation applies directly to how we think about software development, ranging all the way from initial discussions about software requirements through to the maintenance of defective software. Also there is direct relevance of this theory to parts of the work in artificial intelligence, and especially to expert systems work.

A. Gestalt Psychology. Gestalt psychology presses the point of view that each individual possesses a "gestalt" or "world view" that cannot be deduced, elicited, or otherwise discovered simply by knowing what components are involved. The whole is greater than the sum of the parts, thus any process that elicits component parts will inherently be incapable of reconstructing the gestalt from those parts.

[In addition, the gestalt is subject to continuous learning and, possibly, to continuous modification.]

The present practice in developing expert systems ignores gestalt theory, and substitutes another theory that says it is possible to recapture an expert's thought patterns simply by eliciting information from the expert and organizing this information under the analysis of a "knowledge engineer".

B. Constructivism. The modern theory of constructivism holds that each individual builds a personal perception of reality. In this theory, it is never possible to discover "reality". While reality may exist independently of the observer, the observer will never capture it, but instead will construct a personal reality. Obviously this theory bears a very close relation to gestalt psychology.
The constructivist theory holds also that while it is possible to experiment upon parts of the universe, experiments can only deny the validity of various constructs as universals or invariants, and cannot ever affirm the validity.

While physical science has profited most from the assumption of a reality that is observer-independent, even in physical science the so-called "laws" must be adjusted to correct for variations. Attempts to make behavioral science into a model of physical science have failed abysmally. Part of the reason lies in the assumption of a reality that is observer independent, coupled with a belief that observers will have access to such a reality and can discover invariants in this world.

Discrepancies of this type are never more apparent than when one attempts to discuss a new "law" that relates in some way to human behavior. Such discussions invariably evoke extended dialogs about how laws are produced, who is entitled to state them, and under what conditions should they be accepted.*

Practitioners of software development are unlikely to attach much significance to constructivism in its present form. Therefore research has recently been initiated to build a science of perception that will be useful in constructing a theory of intelligent systems. Part of the initial results of this research are presented here.

The Domain of Perspective Model. One of the severe weaknesses in the software field is in management of the software development life cycle. A diagnosis of one of the reasons for this weakness is that it is partly caused by very different perspectives on the part of the programmers and the managers of programmers. In order to see how such differences can be made more specific, consider Figure 1. Here we see a Domain of Perspective model, with two primary branches called "Scope" and "Jurisdiction". Under each primary branch there subdivisions.

*A discussion of these issues is given in the paper by J. N. Warfield titled "A Typology of Laws", presented at the 1986 meeting of the American Society for Cybernetics.
### Domain of Perspective Model

#### Scope

<table>
<thead>
<tr>
<th>Spatial</th>
<th>Temporal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Galactic</td>
<td>Eternity</td>
</tr>
<tr>
<td>Global</td>
<td>Millennium</td>
</tr>
<tr>
<td>Global-Regional</td>
<td>Century</td>
</tr>
<tr>
<td>National</td>
<td>Lifetime</td>
</tr>
<tr>
<td>National-Regional</td>
<td>Decade</td>
</tr>
<tr>
<td>State</td>
<td>Five Years</td>
</tr>
<tr>
<td>State-Regional</td>
<td>One Year</td>
</tr>
<tr>
<td>County</td>
<td>One Quarter</td>
</tr>
<tr>
<td>County-Regional</td>
<td>One Month</td>
</tr>
<tr>
<td>City</td>
<td>One Week</td>
</tr>
<tr>
<td>City-Regional</td>
<td>One Day</td>
</tr>
<tr>
<td>Neighborhood</td>
<td>One hour</td>
</tr>
<tr>
<td>Workplace</td>
<td>The Moment</td>
</tr>
<tr>
<td>Home</td>
<td></td>
</tr>
</tbody>
</table>

#### Jurisdiction

<table>
<thead>
<tr>
<th>Unit of Governance</th>
<th>Perceived Career Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global</td>
<td>Transcendental Organization</td>
</tr>
<tr>
<td>Global-Regional</td>
<td>Multinational Organization</td>
</tr>
<tr>
<td>National</td>
<td>National Organization</td>
</tr>
<tr>
<td>National-Regional</td>
<td>State Organization</td>
</tr>
<tr>
<td>State</td>
<td>Local Organization</td>
</tr>
<tr>
<td>State-Regional</td>
<td>Division of an Organization</td>
</tr>
<tr>
<td>County</td>
<td>Department of an Organization</td>
</tr>
<tr>
<td>County-Regional</td>
<td>Independent Individual</td>
</tr>
<tr>
<td>City</td>
<td></td>
</tr>
<tr>
<td>City-Regional</td>
<td></td>
</tr>
<tr>
<td>Family</td>
<td></td>
</tr>
<tr>
<td>Individual</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 1.** A Domain of Perspective Model is a source from which an individual can develop a relational pattern comprised of one (or more) components from each of the four categories shown. This relational pattern forms a basis for an individual's orientation. The pattern(s) and orientation(s) are components of the individuals constructed gestalt. The individual's Reference Pattern provides or stimulates a Reference Orientation. Those issues that the individual deems salient will have to pass through screens that are informed by the Reference Pattern and Orientation. Once issues are selected, the variety of perceived alternatives for resolution will depend on the Reference Orientation. Criteria for decisionmaking will depend heavily upon the Reference Orientation.
Applying the Domain of Perspective Model. To help clarify the Domain of Perspective (DOP) model, let us use a situation that currently exists in Northern Virginia. This will also let us take advantage of the model to illuminate this situation.

At the present time, there is interest at George Mason University in forming a center to do research on how to make the development and production of software more effective. Such an interest also is found in several other places. In recent years the following programs have been established with this kind of purpose:

- The STARS program of the Department of Defense
- The Carnegie-Mellon funded program
- The Syracuse University funded program
- The Software Productivity Consortium
- The University of Maryland Institute for Advanced Computer Studies

to mention only a few. This selected set is distinctive in that government has been heavily involved either directly or indirectly in starting all of them. The first three are federally-funded, through the Department of Defense. The fourth is funded by a group of corporations, but all of them draw considerable income from the Department of Defense through large software contracts. The fifth is funded by the legislature of the state of Maryland (see Figure 2).

In addition to the foregoing, there are numerous small centers or institutes that have adopted artificial intelligence as a primary area of activity, with much emphasis on expert systems. George Mason University is presently involved in these areas in at least two ways: (a) a proposal to the National Science Foundation submitted by Dean Sage in 1985 and (b) ongoing discussions with the Center for Innovative Technology, which has interests in funding several Virginia institutions to purchase existing AI machinery (Symbolics, Texas Instruments, or other).

The former Governor of Virginia promised the Software Productivity Consortium that $5,000,000 in state funds would be provided to give university support to the Consortium, as an inducement to that body to locate in the state of Virginia, which it did.
THE UNIVERSITY OF MARYLAND

INSTITUTE FOR ADVANCED COMPUTER STUDIES

The University of Maryland Institute for Advanced Computer Studies (UMIACS) was established in January 1985 by Maryland Governor Harry Hughes. UMIACS, while residing on the College Park campus, is intended to serve the entire University of Maryland system as a focal point for research activities in computing.

UMIACS will have ten tenure track research faculty lines for the 1985/86 academic year, in addition to a substantial operating budget to support the research activities of its faculty. Within three years, UMIACS will grow to thirty research faculty lines. Most faculty members in UMIACS will hold joint appointments with academic departments on the campus; their teaching responsibilities would be reduced according to their percentage appointments in UMIACS.

Several faculty positions have been reserved for permanent, full-time appointments in the Institute. These appointments will be 12 month appointments at the senior level and will include commitments from the Institute for research support (e.g., equipment, travel, graduate students, etc.). Applicants for these permanent positions should send a complete resume and names of four references to Prof. Larry S. Davis, Acting Director, University of Maryland Institute for Advanced Computer Studies, College Park, MD 20742. The University of Maryland is an equal opportunity and affirmative action employer.

Figure 2.
Let us now use the DOP model to hypothesize several different possible patterns and orientations that might have primary influence on whatever George Mason University elects to do in developing a software productivity research institute (SPRI). Each of these patterns and orientations can be found as combinations of components from Figure 1.

MODEL 1.
Spatial Scope: STATE
Temporal Scope: ONE YEAR
Unit of Governance: STATE
Perceived Career Unit: DIVISION OF A STATE ORGANIZATION

With these four components of a pattern selected, an orientation can be imagined. Our primary interest is in the state of Virginia, and we are concerned primarily about the next year. We will wait until later to think about the long term, because we are primarily interested in getting moving in a direction that suits the Software Productivity Consortium and meets the funding constraints of the Center for Innovative Technology. This is reasonable, because we are a unit under the state of Virginia and need to be cooperative with the state government and especially the Department of Economic Development, under which the Center for Innovative Technology was formed.

A somewhat different model can be formed as follows:

MODEL 2. Spatial Scope: STATE
Temporal Scope: FIVE YEARS
Unit of Governance: STATE
Perceived Career Unit: DIVISION OF A STATE ORGANIZATION

With these four components of pattern selected, only the temporal scope has been changed. A somewhat different orientation can emerge from this change.

Our primary interest is in the state of Virginia, but we must take a view that is longer than one year. We must consider how the development of our University can best serve this state, while also meeting the most urgent conditions set forth by the Software Productivity Consortium and the Center for Innovative Technology. Also we should factor into our thinking and our personnel considerations the possible impact of getting funded by CIT for AI equipment, and of getting funding from the NSF for a 5-year research center development in this area. We must also recognize
the need to build up our teaching in computer science and information systems, because we will have to teach graduate courses in this area to serve nearby industries; therefore we must not overbalance emphasis on research.

A quite different model can be formed as follows:

MODEL 3. Spatial Scope: GLOBAL
Temporal Scope: DECADE
Unit of Governance: STATE
Perceived Career Unit: TRANSCENDENTAL ORGANIZATION

It is true that we may receive funding from the state of Virginia for starting a software research unit. We owe it to the state, however, to look responsibly at the world situation in software and at the need for our country to maintain leadership in this field. We have learned from the decline of the automobile industry, the near death of the steel industry, and the loss of the electronics industry that we cannot afford to be complacent and close our eyes to the possibility that we may lose the computer industry and all of its related parts such as the software industry and ultimately even the telecommunications industry to foreign competition. It will not serve the state well if we build up an expensive research effort only to find that our results help only other nations, who have taken over the relevant industry.

As a university, we are a kind of transcendental organization. We cannot operate only within the confines of our state government. Instead we must provide leadership in deciding what is best for the state, given its desire to promote the software industry in Virginia. This means that we must be more diligent and thoughtful than competitors who elect to build orientations lower in the DOP model. We must learn what is at the foundations of this software field, develop the missing theory, and create consistent methodology that will help assure national leadership throughout the next decade and beyond.

Recognizing that we must have the active support and interest of corporations who now dominate the domestic scene, we must work hard to build good relations by working with them and honoring their needs, but our first concern must be the larger picture. As a transcendental organization, we are among the unique few who carry the burden of knowledge development and wise application of it on behalf of the society from which we draw our resources.
We must develop and sustain this posture, even though other state-funded organizations may not be able to embrace it and may try to force us to take shorter-term postures that ignore the global picture.

We owe it to the state to try to sensitize other organizations in the state and in industry to see that what we propose is consistent with the best interests of everyone, and puts us in a position to make the kind of unique contribution that is required to stay competitive internationally.

Decision-Making on George Mason University Software Center Planning.

It ought to be very clear by studying Models 1 through 3 above that the criteria for software center planning and decisionmaking will be quite sensitive to whatever orientation is developed. Moreover the management and research program activity will be heavily influenced by the orientation.

These are really behavioral issues, and most likely will not be dealt with well unless a number of the relevant actors begin to share a common orientation—if not to the center at least to the software research domain.

Consistency with the Domain of Science Model.[ ]

Recently it has also been necessary to reconceptualize what science is all about in order to carry on a coherent discussion. All of the foregoing is consistent with this model also. This is valuable, because the Domain of Science Model is likely to be a very valuable framework within which a software research program can be conceptualized and justified.

REFERENCES

D. King, Current practices in software development, Yourdon Press, New York, 1984