SOFTWARE ENGINEERING FOUNDATIONS

A Paradigm for Understanding Software Design Methods

The Future Of Software Process

Software Engineering Technologist
PO Box 4402 Garden Grove CA 92642
palmer@netcom.com

1. INTEGRATED PROCESS DEVELOPMENT AT ROCKWELL

This paper will attempt to survey where we are in terms of the relationship between state-of-the-art and state-of-the-practice in the development and application of software engineering processes to actual software engineering practice. From the vantage point gained through this survey, the essay will attempt to peer into the future of software process development. It will draw upon work done at Rockwell‘s Autonetics Marine and Aircraft Systems Division (AM&ASD) in Anaheim, California, but will emphasize the theoretical aspects of process improvement work.

Certainly the initiatives started by the Software Engineering Institute (SEI) have begun to be felt throughout the portion of the industrial community which responds to Department of Defense needs. At Rockwell AM&ASD we are applying these same SEI driven improvement approaches to all of Engineering. This stemmed from a recent organizational restructuring that moved away from a functional organization toward a team-based organizational structure. Team structures were settled upon as one solution coming out of an organizational excellence exercise. However, we wanted to avoid going from the pitfalls of one organization to the pitfalls of the opposite organization. Thus, when team structuring was applied within the organization, a special process team was set up. The charter of the process team was to focus on and guide the improvement of the product development processes of all the teams within Engineering. Under the auspices of the process team special committees were set up for each Engineering discipline. These committees became recognized centers of process improvement and monitoring. The following committees now exist within our division:

- OSPG Oversight Process Group
- CEPG Cost Estimating Process Group
- SWPG Software Process Group

- HWPG Hardware Process Group
- SAPG Systems Analysis Process Group
- NBPG New Business Process Group
- PSPG Product Support Process Group

This essay will be written from the point of view of the Chair of the Software Engineering Process Group (SWPG). It will address how software process improvement fits into overall organizational improvement. The main point is that the isolation of software engineering from the rest of engineering and the concentration on its improvement without dealing with the overall picture is a fundamental mistake. This only really becomes clear when the whole of the engineering organization is addressed, and the full complexity of the embedding of software within the organization becomes evident. The interfaces of the software process within the overall organization are myriad. If the Engineering process does not treat both sides of those interfaces, then all the improvements gained in software engineering may be easily lost elsewhere. It turns out that many times software is the least of an organization’s problems. In fact, software engineering has garnered so much attention that it is actually more advanced in process understanding than other parts of the organization in many cases. Thus, software process improvement efforts need to act as a catalyst for general organizational process improvement. The tendency to isolate software and to concentrate on improving it as an island in the organization as per SEI direction is a tragedy from an industry viewpoint. SEI needs to broaden its scope to include general process improvement as its charter; or new organizations like SEI need to be formed for a coordinated push to improve all aspects of the organization. For instance, a Hardware Engineering Institute and a Systems Engineering Institute could be formed, and these could work together with SEI to bring about a balanced improvement program.

Our own approach to the problem of process definition and improvement turned out to be very wide indeed. We eventually embraced the goal of producing an Integrated Product Development (IPD) process which spans not only Engineering but also Logistics and Production. In IPD a concurrent engineering team is made up of members from all these functions who work together in an integrated fashion to develop the end product. The IPD team has responsibility for the entire development of the product. We decided that this broad cooperation should be mirrored in our IPD oriented process definition. This is another initiative similar to that of SEI which is directed by our customer. We saw it as complementary to our team-based organizational structure. Thus, the Chairpersons of the different process committees gathered and hammered out a hierarchical decomposition of our IPD process using a dataflow diagram analysis technique. That diagram, once finished, gave a picture of the whole of our IPD process which would be implemented by a team drawn from all parts of our organization. The context bubble of this decomposition is called the Integrated Product Development Process which contains the process building blocks shown in the hierarchy diagram.

At the beginning of the process the software process group identified twenty-five kinds of work needed to carry out software development. Over time we reduced these to a core group of software processes which keep evolving as we understand the problem better:

Software Developmental Processes

- software requirements analysis [stand alone]
- software design [stand alone]
- software implementation [stand alone]
- software integration and testing [stand alone]
- software prototyping [stand alone]

Software Support Processes

- software configuration management [absorbed]
- software review [absorbed]
- software quality assurance [absorbed]
- software documentation [absorbed]

Software Control Processes

- software estimating [absorbed]
- software project planning [absorbed]
- software project monitoring and control [absorbed]
- software external liaison [absorbed]

Software Organizational Processes

- software process improvement [absorbed]
- software technology [absorbed]
- software training [absorbed]
- software parametrics [absorbed]
- software reuse [absorbed]

Of these software processes only four were not absorbed into other like processes shared by other functions within the organization. The processes that were not absorbed were the essential software processes that produce a purely software-related transformation in the product. All other supporting, control or organizational processes were common with systems, hardware and other functions. Within these fifty engineering processes the software process group now owns only seven processes. Within definition there is software requirements analysis. Software prototyping, software design, software implementation, software integration and test are visible as separable elements. Software reuse exists within the System Reuse and Element Reuse building blocks.

There is an important lesson in all of this.

Software is almost completely embedded in the overall organization. Process descriptions for hardware and systems will cause redundancy unless all engineering and other non-engineering functions such as logistics and production are considered at the same time as software process is being defined. The efficiency of defining only one process for all of engineering far outweighs the costs of considering each function separately. Therefore, the SEI is doing us a disservice by pushing us to define software processes and ignoring the fundamental business processes shared by multiple functions. Fortunately, our Division started down the road of defining processes for all of engineering at the same time as embarking on moving up the SEI maturity levels. Hopefully we will be able to reap the benefits of defining one process for all functions. But I wonder how many other organizations will be faced with reworking their processes completely to incorporate the IPD initiatives and to apply them to other functions. The incremental cost of doing all Engineering is about triple. But the reward is that about three quarters of the processes necessary for software engineering are shared with other functions. So costs can be spread wider to make up for increased costs. Also, the major benefit is that interfaces with other disciplines are well defined if all engineering is modeled instead of software in isolation.

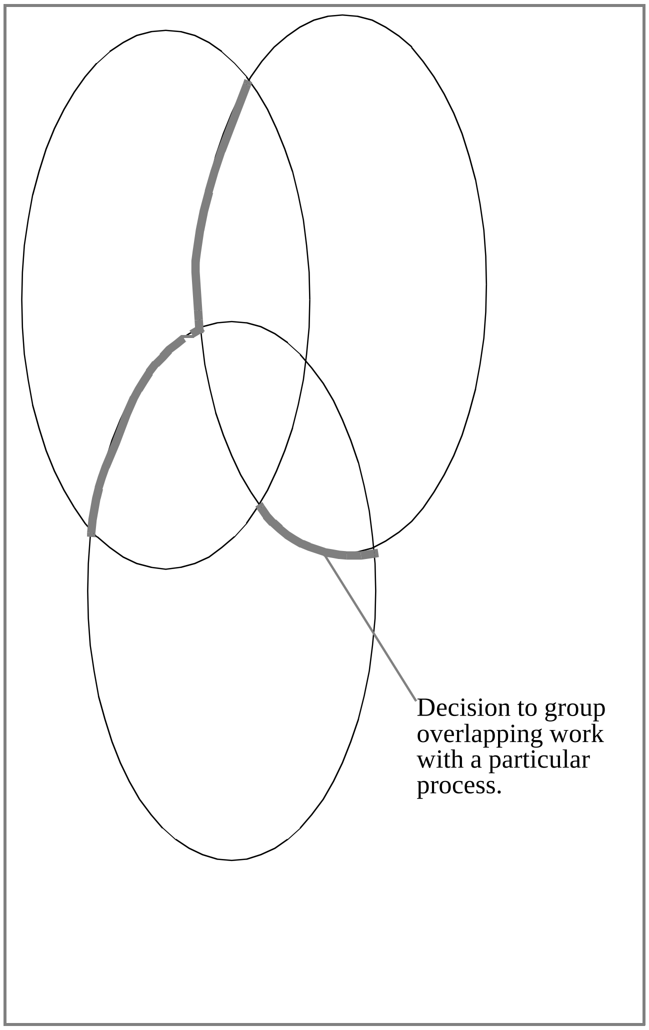
2.THE ART OF DEFINING PROCESSES

Following the example of other organizations, we adopted the dataflow diagram technique as our method for defining IPD processes. This dataflow model was hammered out in a series of meetings of the process Chairs over a six month period. From the beginning we were unhappy with this method of defining processes but did not realize until too late how it should have been done. The important lesson from these meetings was that our organization is so big and complex that no one person knows it all. Also there are many different viewpoints on the organization. Basically these are split between those interested in the global picture of the business and those who are narrowly focused on individual disciplines. The dataflow technique concentrates on the global picture and relegates individual disciplines to the lower reaches of the hierarchy. However, by taking a process building block approach, no element gets slighted because each discipline or viewpoint has its own work that is described by some building block. These building blocks are democratic in that all the different kinds of work that needs to get done must be described somewhere.

The dataflow modeling technique proceeds by functional decomposition which means making distinctions between kinds of work that are subdivided again and again until the simplest task is described. There is an inherent problem with this type of description. Its ability to retain information about the organization of process work is poor. As you go about differentiating kinds of work, it becomes clear that they are all enmeshed. The dataflow technique calls for arbitrary decisions that group kinds of work together no matter how enmeshed they are with other kinds of work. Thus the distinctions are always unsatisfactory. As the distinctions are argued back and forth, criteria are derived which usually, by consensus, sway the group to make the distinction in one way rather than another. However, in our process of deriving the model, we had no obvious way of retaining the basis for the distinction which was drawn. Thus we were constantly feeling as if knowledge was being lost, and many times we would have to re-derive why a particular kind of work was segregated and placed where it was in the hierarchy.

It was not until much later that I realized that the dataflow method, which serves software functional analysis so well, was the totally wrong approach to defining kinds of work. The dataflow technique assumes that from a functional point of view the system being described is not enmeshed but is in some sense homogeneous. Where one makes functional distinctions within a software system is more or less arbitrary because the system itself offers no resistance to functional decomposition. It is generally recognized that there are almost an infinite number of partitions for any software system's functionality. It makes little difference which is chosen since the important distinctions will occur in the architecture, and functional analysis is merely a way to get full and consistent coverage of the requirements of the system. But for sociotechnical work description there is a different problem. Sociotechnical work is totally enmeshed. It is has multiple interpenetrating dimensions. Thus it cannot be easily decomposed functionally because at every level one runs into embedded dimensions of other types of work. For instance, planning occurs in every kind of work to some extent. Even though you have a planning process to produce major planning documents such as the Software Development

Plan, there is still an element of planning embedded in every other kind of work. This multidimensional, embedded and enmeshed character of sociotechnical work is violated by the information retention- poor dataflow technique.



The answer to this problem is to use Venn diagrams with overlapping bubbles instead of segregating hierarchical decompositions. Each bubble can then represent a different dimension. So a planning bubble can intersect all the other process building blocks and still have its own space outside all of these. This overlapping or intersecting of kinds of work represents their multidimensional, embedded and enmeshed character very well. When you draw a set of overlapping circles, it is easy for everyone to understand what the different multifaceted sub-processes are and how they relate to each other. Once all the relations are made clear, then it is easy to make the decision as to how these will be grouped at the higher level. The interior lines of demarcation become dotted for those sub-processes that are grouped with a particular super process. The criteria that allowed this grouping to stand can be recorded as a note, and this knowledge is then not lost. Even though bubbles overlap, data flows between overlapping facets can still be added to the diagram in order to make interface surfaces clear.

It is important to recognize the differences that human sociotechnical systems have in relation to other types of systems and not apply inappropriate modeling techniques when these are known to be inadequate. In this case the multidimensional, embedded, enmeshed character of human work causes pure dataflow modeling to distort the human organization. Because the dataflow assumes the perfect separatability of kinds of work, one constantly runs up against the fact that kinds of work overlap or are multidimensionally related. Using Venn diagrams rather than decomposed bubbles recognizes this inherent character of human work and gives a more accurate model. The point is that because grouping criteria are ultimately derived, one ends up eventually with the same result but with a deeper understanding of the relationships than would be accessible otherwise.

Once building blocks are defined within the context of the hierarchical Venn diagram and the Venn diagram has been converted into a standard dataflow diagram, then it is possible to begin to define processes themselves. In our case we decided to have two basic aspects to our description:

DEFINITION SECTION: WHAT

- Name
- Number

- Dataflow: Showing inputs and outputs graphically.
- Description: Short description of kind of work.
- Inputs: Products needed to do the work.
- Tasks: Lower level kinds of work that make up the process.
- Verification: Questions asked to assure quality of work.
- Outputs: Products produced by work.
- Guidelines: Process related constraints and criteria.

METHOD SECTION: HOW

- Name
- Number
- Tutorial: Explanation of work for novice.
- Procedure: Stepwise ordering of sub-processes.
- Methods & Tools: Standard methods and tools used to do work.
- Metrics: Standard measures applied to work.
- Requirements: External constraints that assure minimal content.

- Policy: Ideal aim of the kind of work.
- Compliance Audit: Required audits of aspects of work.
- Notes: Miscellaneous points such as references.

Every process building block has a document which defines it in terms of a modified ETVX structure and tells how it is to be done. The two sections were separated in order to isolate the parts of the document that we felt would change the most (the how section) and for ease of reference. An exercise was carried out that made sure each of these topics had maximal conceptual separation from each other to reduce redundancy. In addition we listed the inputs and outputs a graphical representation of the process as a bubble with inputs and output lines for quick scan by those who are graphically oriented instead of textually oriented. Descriptions are normally a single paragraph that describe what a particular kind of work entails in general terms.

With respect to the Definition, or What, section we elected to have straightforward inputs and outputs instead of entry and exit criteria. It turned out that the criteria were fairly rare, and we subsumed those under the guidelines section when they occurred. In the guidelines section any process-related guidance as to ordering or concurrency of processes is mentioned which describe how the process is meant to be applied. Also, in the guidelines section the quality goals of consumer processes are identified. These then drive the identification of verification questions which will orient the products of a given process toward what their consumers need. This often is different from what the process itself would seek to optimize if left to its own devices. Thus a crucial connection is made between causal links in the process through the guidelines and verification sections.

The Method, or How, section parallels the What section. It contains a tutorial to explain basic concepts and point the novice to literature which will give him a background in the area. The tutorial should introduce any special vocabulary that will be needed to understand the nature of the process. The Procedure section may be either a diagram showing the interconnection and ordering of tasks mentioned in the Definition section or written step-by-step procedures which tell the order in which the work should be done. Procedures may either be optional or mandatory. Procedures may depart from the ordering of the sub-tasks if necessary. It is in this section that all external requirements are met which dictate a specific action be done. The Methods and Tools sections mention the Division standard methodologies that may be selected from to accomplish the work. The Metrics section mentions the metrics which either may or must be collected on this kind of work. The requirements section lists each external or internal requirement that must be satisfied by the process. These are mapped from the requirements document to each process where they are assigned to lower level processes. In those lower level processes the requirements are listed again, but this time it is mentioned where they are satisfied. If they can be satisfied by a specific action, then those actions are placed in the procedures. If the requirement calls for a state of affairs to be attained or is in some other way unclear then a statement of what aspect of the process that will satisfy that is given. These unclear requirements are marked by a star to make them obvious to the enactor of the process who refers to the document. Requirements state what we are liable to be audited to in the following of our process. Policy, on the other hand, states our goals for a specific kind of work. There is often a wide discrepancy between requirements and policy mainly because our process goals are set high, and it might not be possible to attain those goals in every case. The Compliance Audit is the actual required process review which must be carried out to verify an ongoing aspect of the process. The SEI originally called this a “mechanism” which was a very confusing term. In the CMM they now use the terms monitoring and verification. The Notes section is for the addition of miscellaneous materials not covered in the format which are deemed necessary to add at some later date.

We are now in the process of writing these process definition documents for all of our process building blocks. The same scheme of topics is applied to all levels of process below the process building block level, and our goal is to have definitions of the building block level and at least the next level by the end of the fiscal year. The application of the same set of topics to each level gives an inner coherence to the process description which is elegant and easy to understand once the pattern is learned. We wrote a “process for defining process” using this same format which we gave out to all writers of process definition and which is expected to be followed by them. Thus, we could get a large number of experts working simultaneously on the process definition project. Due to project priorities constantly conflicting with process definition activities, it is a slow and painful process. The expert who is best suited to define a kind of work is also in high demand to do that work on a project. However, once the format is laid down and the kinds of work are isolated, it is straightforward and easy to write these process definitions for individuals with the right experience. It is merely a matter of them setting down and describing their own best practices in the most practical way possible.

The next task is to define the instantiation and tailoring document that shows how these kinds of work are used on a specific project. This is covered in the CMM as Integrated Project Management. We envision using the Spiral Model from SPC as our example lifecycle. For a particular project instantiation and tailoring of the process may be a complex task which needs to take into account the Work Breakdown Structure, Architecture, Accounting Structure, Tailored Standards, Selected Lifecycle, Selected Processes, and many other project specific details. However, generally we see the final set of work packages as specifying which kinds of work will be needed to carry them out. By just specifying the kinds of work, the entire process structure is inherited along with the dataflow interconnections between those processes. Thus we see the use of the defined process as a simple addition to the work already done to plan a project which will bring with it a standardized division- wide process description that makes requirements clear and details the best practices which will allow us to claim to have a defined process to satisfy the requirements of the CMM.

This is an overview of where we are in our attempt to define an Engineering-wide process that embeds software processes but at the same time keeps them visible. In the process of creating this process many deficiencies in the techniques of process definition and enactment have made us wonder about the future of software process. The rest of this paper will address some possible elements of that future and how they might fit together into a complete picture. As is made clear in the remainder of the essay, there are some hard problems that need to be solved for software, and Engineering processes in general, to be effective. The solution to these problems will take some radical rethinking of the software processes and how they are used within the organization. In order to get us started looking at things from a more radical perspective, the following sections of the essay are offered as a point of departure.

3. FACETED AUTOPOIETIC SOFTWARE PROCESS

It has been shown by experimental studies that there is a hierarchy in the effort verses size trade-offs between prototyping, waterfall and spiral process models. Prototyping produces the cheapest and smallest product. Rigorous waterfall adherence produces the next highest cost/effort amount. Barry Boehm attempted to come up with a combination of prototyping and waterfall which he called the Spiral Model. It has been taken up by the Software Productivity Consortium which has produced a complete process model based on this lifecycle. In a recent experiment[1] the spiral model surprisingly produced a larger and more effort-consuming product than both prototyping and the waterfall model. Thus though the spiral model is being touted as the next best lifecycle model, it is actually more expensive than merely following the waterfall life-cycle. It really was an additive combination of waterfall and prototyping.

For software process to work, a new approach to process is needed which attempts to

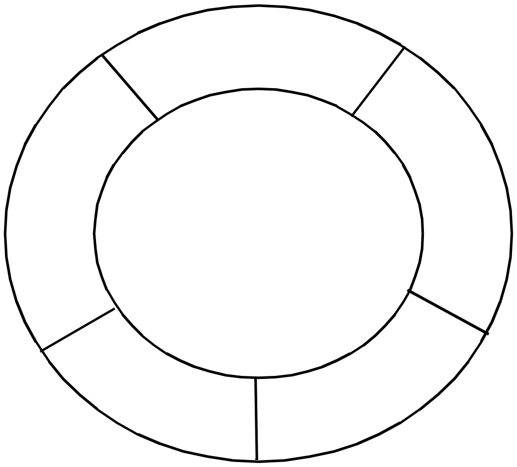
[1] Reference?

approximate the effort/size trade-off of prototyping with added rigor. A new lifecycle model and process model combination is needed to make this happen. The source for this model might be the work done in biology and other subjects on autopoietic systems[2]. Here is a brief sketch of what a software process based on autopoiesis would look like in the spirit of exploring a possible future of process work in general. Autopoiesis means “self-producing” and is generally referred to in terms of self-organization. What we really want in software process is for the software developers to organize themselves so that they produce their products very efficiently and to high standards of quality. Right now the organization that developers left to their own devices imposed on themselves is ad hoc and chaotic. This is an organization which we impose on ourselves, and we could replace it with a different organization. Part of the reason that we do not replace it with a different organization is that all our companies are control oriented so that the practitioners are managed by others who decide the general order and content of the processes that the practitioners will follow in developing

[2] Reference?

products. The problem with this approach is what has been called by R. Ashby "the law of requisite variety" which states that a control channel must be just as complex as the thing it is controlling. Thus, in a control-oriented company a lot of effort is expended to report changes to management and to channel decisions back down to those actually doing the work. This is in fact very inefficient because a lot of information is lost in the transfer to and from the center of control. The other part of the reason is that human beings just naturally produce variety so that each practitioner has his own way of doing things which he defends against all comers. Left to his own devices each software practitioner would do things in his own way. The combination of control orientation and the production of endless variety among practitioners ends up creating a chaotic situation.

Control is always an illusion because of the narrow bandwidth of control channels which cause control to be very gross and prevents refined control past a certain threshold of complexity. There are many pockets in which control cannot be exercised where practitioners are allowed and even encouraged to do things



their own way just so they get the job done and meet the milestones that management cares so much about.

At this point in our software maturity most organizations are imposing a disorganization on themselves which appears to give freedom but is recognized as wasteful if seen in its full implications. Prototyping is merely an excuse for giving the practitioner a means of organizing himself to efficiently produce a product with the minimum interference. The waterfall lifecycle attempts to organize the essential transformations which go on in the practitioner’s head in a way that multiple people can work together toward the production of the necessary products. Thus, in the waterfall transformations multiple intermediary products are produced which can be shared between practitioners working together. In prototyping the larger system would be broken up into pieces that individuals would work on together through controlled interfaces. So prototyping may be seen as a type of cooperation in which the work is broken up into interacting products, but that all stages of the essential transformations remain in a single person’s head for a piece. The waterfall model assumes that multiple people have to share intermediary products for the whole system at each essential transformation of the entire system. These two different approaches are antithetical ways of dividing up the work either as separate interacting longitudinal channels (prototyping), or crosswise with the whole group interacting on each stage in the development of the system.

The Spiral model combines these by doing several iterations of prototyping up front, and then in the end doing the waterfall of essential transformations in the final iteration. Thus it is clear why the Spiral model is an additive combination of prototyping and the waterfall. This combination probably produces a better quality product as all the requirements are surfaced by the prototyping stage up front, yet still the internal knowledge embedded in the product is made visible at the last stage of production. However, just doing longitudinal and then crosswise work organization in two different phases of development does not solve the essential problem of software process. The longitudinal work deployment allows the individual practitioner to generate his own way of doing things but gives control through multiple iterations and the addition of risk management techniques. The waterfall model makes the practitioners settle upon a single way of doing things as imposed by the control channel, but it means that all knowledge must be externalized for sharing in a common format. Externalization of intermediary knowledge is inefficient so that the waterfall is more costly.

Now, let us take a completely different look at the problem using the language of autopoiesis instead of control. Autopoiesis means self-organization. The term was invented by F. G. Varela and H. R. Maturana, Chilean Biologists about 1974[3]. Since then it has

[3] Reference?

caused a small revolution in our understanding of the basic assumptions underlying Systems Theory. It offers a new perspective on the way systems are organized which may help us to understand how to produce efficient processes completely different from the way we are trying to produce them today. In effect, our process work today does not question the control orientation of current organizations. Autopoiesis allows us to question this basic assumption and provides us with an alternative model. So the goal is to have the practitioners organize themselves to do the work in the most efficient and quality-instilling way possible that minimizes unnecessary variety but keeps all essential variety. It must work either in a single practitioner's head or for a group of practitioners working together. It must be a process and lifecycle model that approximates the efficiency of prototyping while adding extra rigor. It must be practically applicable to a wide range of development activities. To accomplish this we will say that software development is not a chain of essential transformational processes, like the waterfall and its variations suggest, but instead is an autopoietic ring. The essential processes in this ring are REQUIREMENTS, DESIGN, IMPLEMENTATION, INTEGRATION, & TEST. Each process in the ring feeds forward and backward to each of the other processes in the ring. The ring can be done by a single practitioner where he is organizing his own activities according to these essential processes or by a team of practitioners. As a team each practitioner may be enacting the whole ring or just a subset of the processes in the ring. The way the ring is assigned to the team, members can approximate a waterfall or prototyping divisions of work. The essential point is that the ring is a consistent set of interlocking processes which are a self-organizing unity. This means that there are certain information flows that move around the ring continuously that are independent of interaction with the outside world. There is a different set of information flows in which each of these processes interact with the outside world. It is the interaction of these two kinds of information flows that needs to be considered very carefully in the construction of a software process model.

In this schema there are four flows of inputs and outputs for each node in the ring. First there is the normal flow which is represented by the waterfall in which requirements leads to design which leads to implementation then integration and test back to requirements. Second there is the reverse flow around the ring in which information is feed back to the prior stages in the waterfall. Third there is the cross ring flow in which each node has inputs from the outside and produces products that are given back outside again. In addition there are also the flows that go from each node to support, control, or organizational processes. This fourth type of flow is depicted in the basic model of the self-organizing ring as the flow to the center of the ring where support processes exits. An important point is that each node in the ring is really identified by the interference pattern set up by the four flows. Each node is a confluence of these flows that sets up chaotic interferences between the information flowing in each direction. One of the reasons that it is so hard to model the software process is that each essential transformation is really effected by all four flows so that the transformation itself is constantly changing its nature depending on the situation as presented from all four flows. This donut is really lopsided from the point of view of routine verses non-routine work. The implementation node is the high point of routineness, while as one moves away from that in both directions, it is toward more and more non-routine kinds of work. A single software engineer will bounce around the ring in his head as he goes through the development process. Variety is created because the transformations are themselves constantly transforming under the multiple changes in all four information flows feeding any particular node in the ring. Thus, when we talk about software process, we are really talking about nodes of interference between flows of information. The interference patterns can be very complex. So the variety in the products by different software engineers are a response to the complexity of these interference patterns and in the meta-transformation of the essential transformations themselves. If we visualize the ring as a set of interference patterns between flows of information, it is easier to see why things are difficult to control and why variety is very high.

Above we have shown the autopoietic software process as a ring with feedback and feed forward, which makes it a torus, and as a knot where each node in the ring is an interference pattern between four different streams of information. We can also visualize the ring to be paradoxical like a Mobius strip with global/ local differences in structure. Like a Mobius strip that locally looks two sided but globally is realized to be one sided, the software process ring is globally continuous but locally discrete. This means that from a global perspective all the different flows form a single overall process while it locally breaks down into different kinds of interrelated work. It is particularly this problem that appears when multiple people attempt to work on the same thing at the same time. Either the people take different parts of the problem and do the whole ring in parallel, or they attempt to work on a section of the ring together. Both of these ways of dividing up the ring are inefficient. One is inefficient because you are attempting to write down and share all the information flowing through the ring. The other is inefficient because there is no global perspective but merely completely different pieces held together by tight interfaces. In effect, what is needed is a way to stack the rings in a meta-ring.

The meta-ring produces an open system

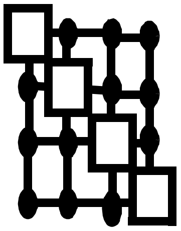
composed of a set of closed systems where a given essential transformation, such as requirements, takes requirements mediated by the whole meta-ring and produces queries for requirements source outside the ring. Or again the design node, for instance, takes technology constraints from the source outside the ring and produces a design consumed by the whole ring. And so on with each of the essential transformations. In such a process the actual work could be divided in either direction. However, whereever the interface is chosen, it must be formalized. Thus, if a different person does all of the requirements nodes in the meta-ring, then an interface needs to be set up with the different designers. Or if several people are doing the sub-rings in parallel, then interfaces need to be set up between each practitioner doing the same essential transformation. Because of the global/local paradoxicality, the meta-rings with its subrings are from one viewpoint a single continuous system while from another it is a diverse set of interference patterns in information flows. Working out interfaces between interference patterns is very difficult.

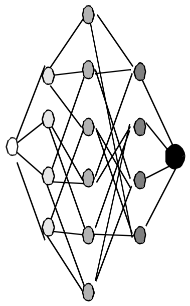
Another completely different way of viewing the rings is as a minimal system[4]. Each minimal system must have at least four independent elements and form a tetrahedron of interrelations. Such a system is really a combinatorial set of all the possible interrelations of facets. As we saw, each node has four different sources of input for information. Thus, the points of the tetrahedron are these different sources. These sources are synthesized into a coherent response at each node so all the combinations of the four must be considered.

In a faceted system all the possible combinations need to be looked at. For example, feedforward needs to be pairwise compared with all three other sources of information (ab, ac, ad). Then all the other sources need to be looked at together without feedforward (a) as in bcd. Finally, all the sources need to be looked at together (abcd). The result of all these faceted looks at the information is the correctly synthesized response. That response is the adapted essential transformation which deals with the particular interference pattern located at that node in the ring. Since each node is producing

[4] B. Fuller, SYNERGETICS Vol1 & 2

a synthesis perhaps concurrently or under recursive or iterative refinement, the entire matrix can have very complex patterns of transformational adaptation and transformational operation. Not only is every node faceted, but the whole ring is faceted.





Thus, for each node in the ring there are several levels of syntheses which will allow the whole ring to achieve unity as a self-organizing system. This synthetic root is the source from which all individual software engineering work is coordinated. Each position in the ring is seen as a gap from the point of view of the rest of the ring (e.g. bcde). In order to fill in that gap it is necessary to confront the interference of autonomous intra-ring information with the external information. That is done by seeing what the pairwise relations between that node and the other nodes in the ring are at any given time. Then the relation between that pairwise relation and the rest of the ring is considered. Thus, the faceted process results in a series of complementary twins which relate the given node with the set of the rest of the nodes or the pairwise relation of a particular node with the rest of the nodes excluded from a pair.

The same faceting occurs between the ring of essential transformations and the support, control, and organizational processes. Thus, the ring is not seen as an isolated unity, but embedded in the organization. Whether we consider these other processes as specific to software engineering or within an engineering-wide IPD framework, all the other processes are shared with other functions and point to its embedding within the organization as a whole. This larger organizational entity may be also an autopoietic unity of greater extent with a set of essential processes that form a ring and subsidiary processes that support, control or organizationally enhance the central autopoietic unity.

Faceted analysis {2^N} is the opposite of relationship analysis. The Lano N^2 chart is the technique of choice for analyzing all possible relationships between elements. This, though, does not allow the interpenetration of the elements to be analyzed. This interpenetration and synthesis of sub-components is a forgotten aspect of analysis which is concerned mostly with how things fit together systems that are assumed to be distinguishable and stable. In Autopoietic systems the inverse becomes an issue because of the embedded nature of human work and the fact that the autopoietic ring is a closed unity or realized synthesis.

We know that software engineering processes are self-organizing because the same essential processes that are applied to producing the software processes can also be applied to defining the software process itself. Thus, the ring of essential processes are recursively entered for process definition. However, we want the practitioners to organize their own processes and thus become truly self-organizing instead of having that process legislated and applied from the outside. The self-organized process may be much more effective and complex than any process applied from the outside. This means that the Mobius strip of product directed essential processes is orthogonally traversed by the same essential processes applied to the process of software development itself. It is the same people working both the process and the product concurrently. If there is only one person, he is both his own process developer as well as a product developer. If there are multiple people, they are all engaged in the same pair of tasks. This means that the process of developing the process grounds the process of developing the product. The same autopoietic ring is grounding itself and organizing itself.

Nothing can be more efficient than this in terms of process enactment. Setting up autopoietic unities that define their own operation as part of their operation is the task of software management. Thus, management must relinquish control and assume the role of catalyst within large organizations. The point is that every individual or team developing software already designs its own process every time. The trick here is to support this activity with guidelines and templates for generic processes, but to stand back and allow necessary variety. Taking the necessary variety out of the process will cause just as many inefficiencies as the ad hoc processes that have a lot of unnecessary variety in them. Finding a middle ground between these two extremes of not enough variety and too much variety is a very difficult task. This task can only be done by the practitioners themselves as a team effort. This application of the essential processes of software engineering to the process before the product is an overhead of operation that will save money in the long run by eliminating inefficiencies through the reduction of unnecessary variety and the enhancement of necessary variety.

4. THE IMPACTS OF THE ASSESSMENT OF SOFTWARE PROCESS

Assessment of currently used actual processes on software projects is a completely different problem than defining ideal processes. In the assessment process one is applying the SEI Capability Maturity Model (CMM) and the associated questionnaire. These are now in draft form, and the 87 questionnaire is still the official instrument. However, we decided to use the CMM and its level two questionnaire as the basis of our own internal assessments. We developed a method which is like an audit in which physical evidence for each ‘yes’ answer is submitted by the project and assessed by the assessment team. The auditors reviewed folders containing supporting materials of each yes answer and marked them compliant or non-compliant. If a project gets all its answers accepted as compliant, then a deeper and certification process is applied. The questionnaire for level two only covers about 38% of the bold points in the CMM. So projects must be certified for each Key Process Area directly. This gives a much more thorough view of the actual compliance status than a normal review that only questions exceptions. However, applying the CMM has brought out several points that are worth adding into our overall look at the software process.

First, the CMM is very thorough and complete and presents a daunting picture for any organization that contemplates compliance. For us the CMM is treated as a requirements document for our process along with DOD-STD-2167A and DOD-STD-2168. Thus, every point within the CMM (goal, commitment, ability, activity, monitor, verification) is a requirement which is embodied in our process somewhere along with every “shall” from 2167A & 2168. This makes numerous requirements that our process has to fulfill along with actually describing the best practice for doing the work involved. Many people make the mistake of treating these requirements documents as the major driver for the formation of their process. This is a mistake because most of the work actually needed to produce the product is ignored by these standards. Any process must first capture the work that needs to be done to produce the products and then backfit these with the requirements from such sources as the CMM.

If full compliance is ever to be achieved, the CMM needs to become a working operations manual for the projects . When it is applied to the actual workings of a project, a good picture of actual practices is obtained. But what the CMM really requires is that one apply the questionnaire, and if it is satisfied, one goes on to apply the CMM itself to assess full compliance. Thus, the CMM becomes the working criteria for how these aspects of a software project are run. It appears to be a very good model which I am sure will improve over time. However, there are some problems that present themselves that need to be addressed:

1) There is an implied organization in the CMM which is at odds with a team-centered organization such as ours.

2) There is no IPD focus in the CMM so that software functions are isolated. Thus, requirements are levied on software that are not recognized by the rest of the engineering organization. This leads to overemphasis on software to the neglect of other functions.

3) There is an emphasis on documentation and verification in the CMM which may be beyond

reason. It is certainly unreasonable to levee such requirements on the software and on no other part of the organization.

4) The fact that there is no process model explicitly called out in the CMM makes compliance more difficult even though it is understandable why SEI attempts to avoid specifying process models. It would help to see a worked example of a process model that embodies the CMM requirements.

Assessment using the CMM and its associated questionnaire makes it clear why a defined process like the one described above is necessary. One clearly sees that many of the problems associated with a project’s compliance could be resolved by having a process description and following it. But the question arises as to how easy is it to enact a defined process. Software is the first industry involving non-routine work to have process descriptions mandated for it. So it quickly becomes relevant to determine whether such a definition and enacting of written processes is appropriate. And here we begin to look at another set of central issues to be addressed in this essay. Can written process descriptions ever be effectively applied to software engineering work, or to engineering work in general? Is this the correct way for us to go about solving our efficiency problems? What are the implications of this for sociotechnical systems in the future?

Assessing one's organization using the CMM as a basis gives tremendous insight into the way one does business. It makes it clear what the benefits of defining processes are in terms of meeting the goals set by the CMM. But it also raises questions as to the direction all these efforts and initiatives are leading which need to be considered from a global perspective. The rest of this essay will approach these issues.

5. SOCIOTECHNICAL SYSTEMS, PROCESS AND NON-ROUTINE WORK

The basic question is whether any person who engages in non-routine work can define their own processes and follow those processes to improve their work. This may seem like a trivial point. But there seems to be no research on this point, and the entire edifice of process definition rests on the ability of a person to follow the proscriptive processes defined by others. Yet if a person who is doing non-routine work cannot describe his own personal processes and then follow them to improve what he is doing, then the entire edifice of process definition and enactment collapses.

In order to approach this question it will be necessary to start from square one and trace the development of process as a way of viewing work in order to anticipate where it is going in the future. The only real process model we have that is published as a standard is IEEE P1074. It identifies software processes as independent of lifecycle. This is the first point that must be appreciated. Process is independent of time. Process is a kind of work that can happen anytime in the lifecycle. What lifecycle you follow is irrelevant. The difference between process and lifecycle follows the analogy of the "set" and "list". Sets are unordered and can contain only one example of any one thing. Lists are ordered and can contain multiple instances of a certain kind of thing. Thus, process models are more basic than lifecycle models. They are partially ordered by dataflow relations and process constraints, but this partial ordering may be violated as processes may begin without all of their inputs or occur in unexpected points in the lifecycle. Processes allow us to see what work is actually going on between project lifecycle milestones. It allows us to see kinds of work that should not normally occur in a particular portion of the lifecycle when they do actually occur. Processes allow us to portray non-routine work as the iterative and recursive entering of processes which are interweaved and enmeshed in the course of actually performing the development of software products. Processes need to be clearly distinguished from a practitioner,s point of view, so it is obvious to him which kind of work he is doing at any one time.

The whole concept of process actually comes from chemical industry plants where statistical quality control methods were applied using sampling. In such a plant there is a physical envelope which liquid ingredients flow down as they are being mixed and transformed. These techniques were taken by Demming[5] to Japan and applied to production lines which have similar well defined boundaries and channels of flow. Now Humphrey[6] and others are attempting to take these same control techniques and apply them to the non-routine work of software engineering. Software development is actually a mixture of routine and highly non-routine work. It is non-routine toward the ends of the lifecycle and routine in the middle of the lifecycle. Each software process has its own mix of routine and non-routine components. Configuration management is mostly routine. Design is highly non-routine. Documentation is about an even mixture. However, never before has non-routine work been submitted to this sort of discipline by attempting to apply statistical methods suited to production lines. The problem is that there is no natural physical channels of flow. The process definition is

[5] Reference?
[6] Reference?

supposed to create ideal channels which will allow measurements. But ideal channels are too malleable, and the changes that defeat measurement are constantly happening.

So the problem becomes whether non-routine work can be treated in a manner similar to routine work from the point of view of process control using idealized or imaginary process containment envelopes which are merely written down on paper. This is not an easy question to answer. The work of Walt Scaachi from USC attempts to deal with this problem. "*The Web of Computing*" article[7] he wrote with Robert Kling of UCI attempted to identify the total environment of computer usage. In that article they borrowed two terms from other researchers which were used to describe "what really goes on." The term "line of work" is used to describe what a person actually does in his job as separate from any job description he might have. The term "going concern" is used to describe the aggregate of the lines of work. These two terms are very valuable as baselines for process description activities. Process definitions are proscriptive or idealized descriptions of what should go on as the norm

[7] Reference?

of rationalized work. But there is always some variance based on individual behavioral patterns and group dynamics from that ideal. The difference between actual and ideal work process needs to be considered very seriously. The only way that process definition can be of any help is if the people actually doing the work can be convinced to ENACT processes. This means that they must understand and follow the processes as described within some limits of acceptability. In Walt Scaachi's work process breakdown is used as a way of understanding the relation between the ideal process and the actual work as carried out by individuals. In this model the process is used to plan activities. The activities are attempted to be carried out. When they encounter obstacles, then these breakdowns are noted and replanning occurs. At this point perhaps the prescribed process is changed, re-instantiated, or re-tailored. Breakdowns are one way of conceiving of the crucial connection between ideal process models and what actually occurs.

Yet is the breakdown model enough? The break down model allows us to adapt the enactment of the process to exigencies of the situation which are unexpected. They assume that our planning is good enough to allow us to carry on a certain distance in the process enactment and that when we encounter breakdowns of our plans, then we will replan and remap our processes to the tasks at hand. This is fine for making process models of routine work adaptable. The question arises as to whether non-routine work is similar to continual breakdown. All the models assume a continuity of a specific process for some amount of time. There is, though, a threshold at which this assumption of continuity itself breaks down. In non-routine work there may be an actual fusion of different types of work in which the hopping from one to the other is so quick that there is no actual dwelling in a particular process. The ultimate, is when one is actually performing different kinds of work in one's head, moving from one to the other is merely the movement between different links in the chains of ideas. This fusion of processes causes boundaries to disappear between types of work and planning to become continuous adaptive behavior which never ceases.

If non-routine work is actually like a continual breakdown state, then there is no way to raise it from the initial level of process maturity. We might characterize this positively as the spontaneous dealing with problems and issues on the fly without time taken to actually work the different issues which inform the problem at hand. A lot of work done by experts and executives is of this type. It is not done in a coherent rational way, but instead is done spontaneously by intuition of the moment or by applying a set of heuristic rules learned by experience. It is based on skill and experience, not on any routinized set of activities. The expert or executive is self-organizing in the sense that what is needed at the moment is produced to the requirements of the present situation. If the circumstances allow, the expert or executive redefines the situation in order to give the right answer for the fulfillment of the perceived need of the moment. Thus, the expert or the executive has the power in many instances to completely redefine the work to be done in accordance with the means of accomplishing that work and prior formulations of that work. Thus, the positive side of continual breakdown is the possibility of creative redefinition of the whole situation in a way completely different from the work’s original formulation. By such a leap the person engaged in non-routine work can change the rules and assumptions that initially defined the work in order to transform what needs to be done. This creative side of working is not covered by any current models of process. However, potentially it is through innovation, creativity, and other forms of non-definable behavior that the greatest gains are to be made in the solution of our problems.

Any definition of process that does not take into account the action of creativity and innovation is flawed. The breakdown metaphor covers it by implication but not explicitly. In fact, what we wish could happen is that all aspects of our process could be the focus of the creativity of our engineers. We would hope that the process is really what Thomas Kuhn called “normal science”[8] between the paradigm changes which redefine the whole field of software work. In normal science we work out the implications of a way of seeing things. But paradigm changes precipitate changes to the whole way of defining the work to be done. The point is that some people in the organization are continuously redefining the work to be done. These experts and executives are involved in

[8] See Structure of Scientific Revolutions

highly non-routine work which cannot be captured by any process model. This creative center of instability is the epicenter of the process. As we back away from it, we find that there are degrees to which every worker can redefine his work. The problem becomes getting others to buy into his redefinition. This process of negotiation causes the whole field of work to constantly change. But it does not change a piece at a time. Instead, the whole pattern of work and its actual fabric is continually being re-patterned and renewed. Work is a gestalt which is constantly changing in spurts with many starts and stops.

Now this means that simultaneously the ideal patterning of the work is changing along with the “lines of work” within the “going concern.” The work environment can handle this because of the essential freedom of the individuals to redefine their work and respond in real time to the re-definitions of work by others. This aspect of non-routine work can never be captured by process. In fact, process will always be trying to keep up to these changes. The minute you write down a process it is obsolete. To the extent people are not engaged in the creative transformation of their work is exactly the extent that process definitions hold long enough to be applied. Thus, process is in constant tension with the creativity of the individuals who are constructing their world and the work within their world spontaneously as they go. From this view everyone is improvising all the time. The making up of the script as it is being performed has always been recognized as a very important aspect of the preparation of the acting profession. In process enactment it is no different. The enactors are defining the process as they go along. Sometimes they can rely on prepared scripts, and sometimes they must rewrite the script on the spur of the moment or even ad lib. In this sense they are self-organizing, autopoietic, systems. They are producing themselves, projecting what they should be and realizing it simultaneously.

Now there is obviously a spectrum from the extreme non-routine work of self-definition to more mundane types of non-routine work in which known and stable processes are continuously being re-entered iteratively and recursively based on the judgment of the expert practitioner or on interrupts from external sources. However, on the whole people know that they do not want to write work proscriptions down because they are just spending extra time producing something that is obsolete as soon as it is produced. Whatever is captured must be both very important and stable for that to be worthwhile. We are trapped by the fact that writing things down is the only way we have to capture information. What we really need is some intermediary form between action and writing which allows us to record our processes without formulating them into sentences because writing things down is an essentially different kind of activity from doing the action. It is only if the representation actually helps one do the work that it is worth this trouble. One half-way house that exists between writing and action is graphics. If there were some way of diagramming process enactments that record the essence of the enactment without having to write descriptions, then these would be useful. However, they would have to actually help the practitioner improve his process performance. Another half-way house that exists is brief notes. If there was a way to describe process which people can keep in note form which is helpful in the enaction of processes, then this may be a way to enable process enactment that makes a bridge between lines of work and proscriptive processes.

Taking the second line of approach to finding a half-way house, I decided that I should perform an experiment in which I attempted to see if a person can record his own process and then follow it. I looked at the planners that are carried by people to help them organize their time, and asked if there was any similar system that could be used for people to apply process to their own work. In this vane I came up with the idea of personal process as the intermediary stage between ideal proscriptive processes and lines of work. The personal process is a way for each individual to describe the kinds of work he engages in each day and attempts to improve that work. The person is spurred to continuously ask himself if this work he is doing can be redefined or reorganized so as to become simpler or eliminated altogether. If the work cannot be redefined, then one is asked to define each kind of work one does and manage the kinds of work as a complete pattern. If the kinds of work are highly non-routine, then the iterative, recursive, or interrupt driven pattern of process enactment is described. If the work is routine, then the sequential pattern of work is defined instead. In either case the personal process can set aside specific time for considering redefinition of work, defining work processes, managing the pattern of personal work processes, or managing work assignments.

6. PERSONAL WORK PROCESS MANAGEMENT

My experiment with personal process management was conceived to see if process description and enactment is practical. I entertained grave doubts as to the efficacy of process work within organizations. But I hypothesized that if personal process management can be made as useful as time management, then it could actually have some impact within real work settings. Personal process management exists at an intermediate level between the proscriptive work process and the individual's line of work. It is owned completely by the individual, and is a means of controlling and rendering his non-routine work more efficient. It is modeled after the time planning books that are used almost universally to manage coordination and action assignments by individuals. Personal Process Management is an extension of this system which attempts to give the individual control over their entire work pattern.

The steps of personal process management are fairly simple.

1) Describe the kinds of work you do on the process description sheet. This sheet has areas for notes about inputs, outputs, steps, issues, notes, problems, breakdowns, and the normal phases of any process.

2) Describe the pattern of work processes and how these interrelate, including the routine or non-routine nature of these interactions between processes on the control sheet.

3) Describe the work assignments and action items associated with those work assignments as they appear and relate them to the processes.

4) Consider how the work assignments could be redefined in order to increase efficiency. Do not accept a work assignment without considering how it could be redefined.

5) Consider how action items relate to processes, and again test each action item for the possibility of redefinition.

6) Negotiate redefined action items and work assignments with their organizational source in order to make sure that there is common

understanding of the new work to be done.

7) Consider the work source interface carefully. If the work is coming from the work source in some form which is difficult to act upon, then attempt to clarify the inputs, or get redefinition of those inputs.

8) Consider the whole pattern of work to see if it can be streamlined and made more efficient.

9) Consider each personal process within the whole pattern of work, and attempt to make each of these efficient.

10) Attempt to apply personal measures to the work assignments flowing though the processes in order to have a basis for improvement of those assignments.

11) Consider interfaces with the personal processes of others. Model the processes of others that you depend on, and attempt to get them to make changes that will increase overall efficiency of the team.

12) Consider the whole work process within its organizational and proscriptive process

setting. Attempt to change the organizational structure and proscriptive process in order to improve the overall efficiency.

Now what these steps show is that work redefinition is the center of personal process. Proscriptive processes must be written in such a way as to allow tremendous variability but still supply standard non-changing definitions everyone can use for a prolonged period of time. Yet personal process represents the “in the trenches” tailoring of proscriptive processes based on the exigencies of the situation. When you start recording your personal process, it is clear that inputs are mostly implicit when any work assignment is given. Those that are explicit are normally things you have to construct yourself or find. The number of explicit handed-over inputs are usually very few, for non-routine work outputs must be made up. They will probably be different for each work assignment, and the steps done will change on a case by case basis. Yet, by thinking through the processes one is doing, it makes them clear and allows one to sort though the recurring problems. This exercise is valuable in itself and is actually what will improve the process overall. We need to think about the processes we perform instead of blindly executing them. In American business culture there is a lack of emphasis on thinking about what we are doing. When it is mentioned, it comes under the rubric of working smart. We need to motivate individuals to think through what they are doing, and personal process analysis facilitates this. It lets the individual step back from what he is doing and get a global perspective based on a record of actual instances focused on the kinds of work he is doing now. The kinds of work people do who engage in non-routine work do will vary widely from month to month. They are called upon to do the hard jobs which demand creativity and intensity of execution. But for a given period the work to be done will fall into a pattern which is amenable to personal process analysis.

The analysis is performed as an ongoing reflection on the kinds of work being engaged in, not as a one-time snapshot. It allows the person to say to him/herself:

- What are the main kinds of work I am engaged in these days?

- How are these related to eachother in an overall pattern?
- How could I redefine this work to simplify it and better integrate it with the work of others?
- What are the issues arising in my work assignments for a particular kind of work?
- What are the problems I am having to deal with over and over?
- What are the points where my work assignment plans are breaking down?

The elements of personal process analysis form an integrated system of elements which needs to itself be dealt with in a systematic way.

- Work assignments have tasks.
- Work assignments have priorities.
- Work assignments have resources.
- Work assignments have a due date.
- Work assignments have specified outputs.
- Tasks have action items.
- Tasks embody personal processes.

- Personal processes are the basic pattern of work expected by a series of similar work assignments.
- Work assignments may be self-assignments or externally elicited.
- Personal processes are kinds of work distinguished by the individual.
- Personal processes have inputs which may be explicit, implicit, given or to be found.
- Processes have steps by which the work is normally structured.
- There should be some isomorphism between action items and steps at some level of process definition.
- Working action items through process steps produces issues that need to be resolved.
- Resolution of issues may take deliberation with others.
- Any issue that cannot be resolved within the scope of resources allotted is a risk that must be channeled to the person responsible for risk assessment, avoidance, and abatement.

- Processes when executing a specific work assignment by executing its process steps by action items may encounter problems generic to all similar work assignments.
- Problems when identified must be worked separately from the action items associated with the problem.
- Problems may demand expert assistance.
- Problems are knots of immediately unresolvable issues.
- These knots of issues may generate risks if they go unresolved too long.
- Problems may be wicked in which case there is no optimal solution, wicked solutions require a trade off.
- If enough wicked problems occur, there is a breakdown of the process.
- Breakdowns call for replanning and perhaps redefinition of the work to be done.
- The generic stages of process execution are as follows.
- Formalism selection and validation.
- Exploration based on using the formalism on given content.

- Elaboration once formalism is seen to work.
- Assessment which calls for re-evaluation of issues, risks, & problems.
- Verification that results are actually still true and that changes elsewhere in the system have not rendered them obsolete.
- Inference considers the effects of my results on other work assignments.
- Consistency checking of results.
- Evaluation of results in light of requirements for outputs.
- Deliberation on issues with others.
- Validation by application of available knownledge and criteria to the results of the process.
- Invocation of other processes to deal with outputs.
- Context switches from one process to another cause lags that require time to reacquire the work situation.
- Interrupts cause context switches.
- Personal processes may be entered iteratively or recursively.

- Personal processes may be entered at a particular level of detail or with a specific kind of input which causes different responses.
- Personal processes may instantiate prescribed processes defined generically by the organization.
- Personal processes create an imaginary envelope around segments of an individual’s line of work.
- Personal processes allow an individual to take control of his own work and give him leverage for redefining his work within the organizational context.
- Personal processes revolve around work redefinition and attempt to maximize the efficiency of work and eliminate unnecessary work.

These entity relationship definition propositions define the arena for personal process analysis. Making process a conscious activity for the individual wherein he makes “objective” certain aspects of his line of work is the key to overall process improvement. Once such an analysis has been performed, sampling from his own ongoing work stream, then it is possible for the individual to relate that analysis to other dimensions of his working world, such as the political, social, organizational, and facilities dimension. Each of these dimensions are incomplete and under construction. An individual's working life is multidimensional, and once a picture exists of the work stream, it is possible to use it in many ways to improve the working environment of the individual.

My experiment with personal process showed me that it is really difficult to define processes and enact them. These results have strengthened my skepticism regarding the ability of companies to define processes and legislate them on people. The major problems I encountered were as follows:

- It is very difficult in the "heat of battle," so to speak, to sit down and define one's processes. What we are really talking about here is thinking about what you are doing. For the most part people do not think through their actions and their implications.
- Even if you force yourself to sit down, it is difficult to isolate kinds of work. It

seems that isolatable kinds of work are exceptions rather than the rule. Many times work comes in snippets where one is doing something in an one-off fashion. Or one is called to improvise on the spot. So work appears more fragmentary than the process model of work allows.

- Another problem is that work is almost entirely, in many cases, interrupt driven. The process model does not account for interrupt-driven work. What is needed is some sort of context-saving mechanism for work which allows one to reacquire a particular context more quickly.
- We are all so product driven that the shape of the product almost completely defines the nature of the work to be done. We guide our work by borrowed templates, or worked examples, to a great degree, but unfortunately most of these templates either do not exist or are too different from what is needed in a particular case to be useful. Thus, because the work we do is so product specific, working out a process is

redundant in many cases. Processes are by nature generic. Unless the work that is done has some generic elements, it is not clear how processes will be adapted to them.

- Processes in many instances have no inputs. If we are given an assignment, then we must get the inputs needed or invent them ourselves. This means that work does not flow in a dataflow fashion. Whether this is just a symptom of chaotic processes is not clear. Whether dataflows can be set up in human organizations is not clear.
- Many times the person that receives the data produced from a process does not react. Thus, there is the phenomena of output going nowhere. It exists only because one’s boss asked for it, not because anyone else in the organization really needs it. People in the organization do not know each other’s real needs. They are kept isolated by organizational structure. Many times it appears that a control-oriented organization cannot have a process because so much time is spent creating

and maintaining organizational barriers, that flows between organizational segments is precluded by necessity.

- What seems most useful is when two people discuss what the two will do together. I will do this, and you do that ,and we will produce such and such. It seems that personal process has the most to offer in this context because people actually do get together and outline mutual responsibilities. But this stage is usually not dwelt on, and many times is circumvented by control-oriented structuring of the work by management.

The fact that creating personal process is hard suggests that perhaps the context in which it is done is important. In control-oriented organizations it is clear that one’s influence on the way things are done is limited, so it is difficult to justify doing much to optimize one’s own work. This is especially true if no one else is attempting to make the same kinds of optimizations. Thus, it appears that autopoietic systems are difficult to create and maintain by oneself. It needs the whole group to participate, and there is really no call for doing this in a control-centered organization. In fact, it is counter to the flow of work set up by power relations in the organization. So the question arises as to how the autopoietic process can differentiate itself in order to become established. Unless the autopoietic system has some basis other than in individual process actions, it can never be put into action. This really says that the autopoietic system needs to have an internal differentiation which is based on something other than just arbitrary distinctions. It needs to have its own naturally arising intersubjectively valid structure which will provide a different context for personal process management.

7. GENESIS AND MODELING OF AUTOPOIETIC PROCESSES

A question that should be addressed is the origin of the autopoietic ring of software engineering essential processes, and once that is understood, we should consider how to model them. It is one thing to say we need to arrange our processes in an autopoietic formation, which is self-organizing, in which people apply personal processes and proscriptive processes to do their work, and quite another to be able to have a formal model of these processes. Industry needs a model which in both proscriptive and can be enacted creatively to produce the requisite variety. That process must be amenable to being modeled. The real challenge of the future is to get these different dimensions correctly balanced in our process models. This necessitates an understanding of the nature of software itself and the unique aspects of the software engineering discipline and its relation to general systems theory.

Here we will provide a short synopsis of work on the nature of software and the relation of software to general systems theory. Software is seen as a new kind of object based on a completely different foundation than other entities we normally encounter in the world. Briefly there is a hierarchy of levels of Being. Software falls at the third meta-level of this hierarchy in a type of Being called by Merleau-Ponty Hyper-Being. The first type of Being that is below this level at which software falls is Pure Presence which is the type of being that applies to everything that can be the focus of our attention in the world. This is the kind of being defined by Aristotle, elaborated by Descartes, and finally formulated by Kant. The second type of Being is called Process Being which is the level of all technology. It is dynamic rather than static like Process Being, and we do not see it explicitly except as technology breaks down and our attention is drawn to the underlying technological infrastructure that supports our activities which are focused on some purely present object of our intentions. Hyper-Being was first identified by Heidegger and later elaborated on by Merleau-Ponty and Jacques Derrida. It is defined as the inner coherence of the technological system which never appears. Michael Henry identified it with what he calls the Essence of Manifestation in his book of the same name. It is like the unconscious of the technological system and like Freud’s unconscious causes distortions in the showing and hiding of manifestation without ever being seen itself. Derrida describes these distortions in terms of what he calls DifferAnce which is the differing and differing of texts. Texts are a field of pure difference which allow us to manipulate them outside the flow of time. In other words, what is written first may be placed last in the sequence of the text, so the display of text allows a different order from its production, unlike speech. Software is a kind of animated text which has all the aspects which Derrida identifies for text, and thus operates at this Hyper-Being level of manifestation. Beyond Hyper-Being is a final level of Being, or level of describing manifestation, which is called Wild Being by Merleau Ponty and which was positively described by Deleuze and Guattari in their books on Capitalism and Schizophrenia. This final level of Being is that at which artificial intelligence operates. It uses software as a virtual machine the way software uses hardware. Computer hardware embodies the first two meta-levels of Being where software embodies the third layer and artificial intelligence techniques embody the last highest level.

This quick tour of the ontological basis of software is meant only to establish that software is a new kind of entity which is based on a kind of Being which is different from what we are normally used to dealing with in our world. Thus, software and its production processes have attributes that are different from other kinds of objects such as hardware. This new kind of entity must be understood in terms of a different paradigm that is adapted to its mode of manifestation in the world. Attempting to understand it as if it were hardware will always fail. One of the salient differences is that software is a theoretical object and exists as an abstraction. These abstractions are described by software methodologies. An analysis of these methodologies has led to a description of the field of all possible methods. This field is comprised of four unique viewpoints on software theory: AGENT, FUNCTION, EVENT, and DATA. The interrelations between these viewpoints are the basis for software methods. Each viewpoint has its own specific software method, plus there is one minimal method for each one-way bridge between viewpoints. This gives us a set of sixteen software minimal methods that describe the entire field of possible software methods. These individual methods together form a system of representations that approximate the form of the software theory. Not all the aspects can be seen at once, but there is always some feature of the software theory that is hidden. The inner coherence of the system of viewpoints and their associated methods is the always hidden essence of manifestation, or the unconscious of the technical system. It only appears with the introduction of software into the technical system which at once gives the possibility of integrating various aspects of the technical system into an autonomous whole, and at the same time makes the technical system as a whole unavailable to complete scrutiny. If nothing else, the technical system as it executes has compiled software modules which cannot be seen into and remain in their compiled version totally opaque. This opaque aspect of the software bound together technical system has been described elsewhere by the author as the Meta-technology which has its own emergent aspects not encompassed by traditional views of technology.

An important discovery has been that the technical infrastructure that provides the four viewpoints is only coordinated, by the appearance of methodologies at the meta-technical level. At the technical level these four views are not coordinated and also the non-manifesting aspects of the technical system do not appear. This relates to the fact that the viewpoints themselves have different inherent ordering properties that only are recognized when they are brought together as a coordinated meta-technical system. In fact, it is seen that the AGENT and FUNCTION viewpoints are only partially ordered in relation to the EVENT and DATA viewpoints, that represent the spacetime continuum, which are fully ordered. It turns out that the minimal methods are built on the intermediary positions between these two kinds of orderings which is “Linear order without distance” and “Partial order with distance.” The only other type of ordering is the non-ordering of pure distinctions. The fact that all minimal methods are duals which are based on these two types of intermediary orderings explains a great deal about the structure of the minimal methods. This connection between the ordering of the viewpoints allows a coherent connection to be made between the General Systems Theory of George Klir[9] and the software methods. In that extended general systems theory any software system may be modeled. But further, it represents a general set of modeling techniques by which any discrete dynamic system may be modeled.

This is an important point because these same modeling techniques that are used to model the software theories that are implemented into software products can be used to model the software process itself. That modeling needs to be augmented by some continuous modeling techniques, but all discrete aspects of the software process can be captured by the exact same techniques for modeling software theories. This reinforces Osterwiel's point that doing software process work is analogous to building the software product[10]. This recursive nature of software in which the methods for describing the products and processes are the same is another indicator that software is a different kind of entity that immediately produces paradoxes and recursive self-referential structures.

[9] Reference?
[10] Reference?

It is possible to extend these insights into the nature of software design to generate the software autopoietic ring of software essential processes out of what we know of the software methods and their viewpoints. We do this by realizing that software design has four fundamental viewpoints. But there is another viewpoint available which has not been used in describing software design. This other viewpoint is that which is described as the source of unordered distinctions. We will call that viewpoint the Catalyst. The Catalyst viewpoint is the source of a never ending variety of distinctions and discriminations which relates the software design theory to the world. This relation of the software design theory to the world is the source of the autopoietic ring of essential software processes. Basically this is done by substitution. In the other phases one of the fundamental design viewpoints is hidden, and the Catalyst viewpoint that makes unordered distinctions is substituted. This generates five tetrahedral structures out of the single tetrahedron of design viewpoints. This also connects the abstract design to the world because the Catalyst viewpoint makes concrete practical distinctions which constrain the design.

REQUIRMENTS

- Catalyst viewpoint (4) (Agent hidden)
- Function viewpoint (2)
- Event viewpoint (3)
- Data viewpoint (1)
- This agrees completely with all known requirements analysis methods which are unanimous that the agent viewpoint is suppressed in this essential process. The Catalyst viewpoint is the source of requirements distinctions which become performance thresholds or functional demarcations which are represented as “***shalls***.”

DESIGN

- Agent viewpoint (1)
- Function viewpoint (3)
- Event viewpoint (4)
- Data viewpoint (2)
- The design viewpoint is purely theoretical, and so the Catalyst viewpoint is itself suppressed.

IMPLEMENTATION

- Agent viewpoint (2)
- Function viewpoint (4)
- Catalyst viewpoint (1) (Event hidden)
- Data viewpoint (3)
- In implementation the Catalyst viewpoint provides the pure distinctions between 0s & 1s to which the software ultimately reduces. The source code itself is purely data, and even the compiled code is just data. The implementation deals with the programming language and the embodiment of the design elements in software modules which deal with the effects of delocalization which smears the design elements out in the actual code. This effect of delocalization is another way in which the catalyst viewpoint appears within implementation.

INTEGRATION

- Agent viewpoint (3)
- Catalyst viewpoint (2) (Function hidden)
- Event viewpoint (1)
- Data viewpoint (4)

- In the integration the functional viewpoint is suppressed. The integrator does not care about the functionality of the pieces he is assembling, but only if the components fit together with each other and the hardware environment properly. The Catalyst viewpoint discriminates all the points of proper connection and improper disconnection between pieces and the hardware substrate.

TEST

- Agent viewpoint (4)
- Function viewpoint (1)
- Event viewpoint (2)
- Catalyst viewpoint (3) (Data hidden)
- In test the data viewpoint is hidden. The Catalyst viewpoint discriminated between correct and incorrect performance on tests. The test data is only traces of events that occur when the test software interacts in execution with the product software. Since we continuously think about test data, it is strange to realize that data is really only records of events that occur during

> testing and that the event perspective is the crucial one to be considered in testing proper.

In this picture of the addition of the Catalyst viewpoint to four fundamental viewpoints on design, we have managed to produce the autopoietic ring of essential software processes. This is an important step in understanding the structure of the ring itself. In the ring there is a differentiation of key transformations of the software product. This differentiation arises when software is done by more than one person. When one person does software by himself, it is called programming. It centers around the production of source code which runs. It only has to satisfy its producer. Many times this isolated programming is called "hacking." Hacking suggests trial and error ad hoc processes. It also suggests working something over and over until it works. The hacking process by a single individual has the different essential transformations fused. The requirements are not distinguished. The design is embedded in the code. The code is perhaps a monolithic mass of highly concentrated interactions analogous to spaghetti. The code is probably not specifically tested; it is just run until it fails. Since design is not distinguished, there is no real implementation process which specifically deals with delocalization. Everything takes place under the shadow of delocalization. Thus, all the worst effects of what Derrida calls DifferAnce appear in the hacking process. If hacking is carried over into software developed by more than one person, then these effects escalate exponentially to cause severe problems. In other words, hacking is good for creating computer viruses that you don’t want anyone else to know about; but hacking becomes counter productive when just one other person is added to the project either as customer, co-producer, or operator. The autopoietic ring of essential processes appear whenever two people attempt to develop software together, whether they divide the work so each does a separate part in a prototyping mode, or whether they do each essential process together as the waterfall model suggests. This means the autopoietic ring is an intersubjective phenomena. In that cooperative situation there is a natural differentiation of work which allows cooperation. The autopoietic ring appears as this natural differentiation of the fused hacking process into separate essential kinds of work. It is possible to relate this to the appearance in linguistics of MOODS or universal human transactions. We can identify the following five basic transactions:

- COMMAND -- Requirements

These are commands that the system SHALL do such and such.

- PROMISE -- Design

The Design explores the different possibilities of the design landscape and searches for the most promising or optimal design

- STATEMENT -- Implementation

Implementation results in a series of source statements.

- NEGATION -- Integration

The different pieces when put together may negate each other, and this contrariness must be overcome to fit the system together stage by stage.

- QUESTION -- Test

The builds of the system are tested by running it against test cases and test software which question the software as to its embodiment of the requirements.

This set of Moods is not undisputed; different linguists have their own lists. But all lists contain at least Command, Statement, and Question. But these Moods may be considered a representative set. They do not have any force that we might infer if they were the only possible transactions attested in all known languages. But Moods are right on the borderline between grammar and discourse, which is the distinction Saussaire made between Language and Speaking[11]. They represent that area of language where the structure of the sentence effected the actual speech transaction you are having with someone. This is interesting because it is exactly where the autopoietic ring comes into existence. The fact that we can see a pramatic dialogic structure which is embedded in grammar similar in many ways to the qualities of the phases of software development is intriguing. It allows us to connect the qualitative differentiation of the software

[11] Refereence?

essential processes to something we all understand and use every day which is different moods in our discourse. It is interesting to attempt to think of a world in which there are no questions, or a world without commands, or without some of the other moods. This is an exercise that allows one to see that these moods are indeed crucial to our existence. If we could not ask questions, our ability to learn would be severely constrained. And we would definitely have no notion of testing software. If we could not give and receive commands, then our ability to coordinate with others would be curtailed, and we would also have a hard time coordinating our software development with the needs of others, as we would have no notion of requirements. And so it is with the other moods which exemplify essential transformations of software, until we realize that each of the essential transformations exemplifies some aspect of the intersubjective process of working together which depend on human cooperative faculties that are deeply embedded in our beings. These cooperative faculties, such as the ability to transmit information as statements between eachother, or the ability to make promises to eachother, form a ready-to-hand part of our being-in-the-world with others. This manifests itself in certain linguistic phenomena, but also it manifests itself in the natural structuring of the cooperative work process that has become legitimized across many disciplines.

What is unique to software here is that this basic sequence of steps in software becomes an autopoietic ring instead of a linear process with beginning and end. In software the snake has eaten its own tail in many ways. First, the essential processes form a ring which must be broken to stop. Software products keep evolving. They are not etched into hardware and may continue to change in ways the hardware could never change. The reprogramming of Voyager to go on to explore more planets, one after the other, is a case in point. The adaptations of the Voyager spacecraft were made in flight by changing the software, where no hardware changes were possible, other than those caused by deterioration of the platform. Thus, software once tested is merely the basis for a new set of requirements which cause it to be totally revamped. We now artificially separate first time development from maintenance, but actually this is all one process in which the autopoietic ring once established is self feeding. Beyond that there is the fact that software essential steps are the same steps that must be undertaken for the software practitioners to organize their own work. Thus, within software developing the process goes hand in hand with development of the product. In this the autopoietic ring is applied to itself in a way that is unique to this production process. In all other types of production processes it would be absurd for anyone to say that process is software too. But in software it is difficult to distinguish between the product and the organization of the work to produce the product.

Once we accept the augmentation of the four fundamental viewpoints on design with the Catalyst viewpoint, and the fact that this generates the differentiation of the autopoietic ring, then we can go further and look into what this tells us about the autopoietic ring itself. For one thing it means that there is a whole new set of minimal methods that arise which represent the relation of this new viewpoint to the other design viewpoints. In this case the new minimal methods will have a different set of characteristics because they relate to the generation of distinctions within the context of all possible distinctions and relations between things.

- Catalyst alone Logic of Form

G. Spencer Brown[12] and Francisco Varela[13] have developed a logic based on the simple act of making a distinction. It is the prototype for all distinctions that are made by the Catalyst.

- Catalyst >>> Function Lano N^2 [14]

All possible relations between functions.

The Lano N2 chart allows all possible relations between functions to be made visible for analysis.

- Function >>> Catalyst Facet Analysis 2^N

All possible functional decompositions.

Faceted Analysis and Synthesis allows all possible emeshings of different decompositions of the same thing to be studied.

[12] See LAWS OF FORM

[13] Reference?

[14] Reference?

- Catalyst >>> Event Temporal Logic[15]

All possible event sequences.

Temporal logics studies the relations between necessary and variable event sequences to be studied.

- Event >>> Catalyst Combinatorics[16]

All possible relations between events.

Combinatorics allows the permutation of event sequences to study all possible sequences.

- Catalyst >>> Data Information Theory[17]

All possible relations between data elements.

Information theory studies the coding of data, usually for transmission, but newer forms of information theory consider the building up of patterns of relations between pieces of information.

[15] Reference?

[16] Reference?

[17] Reference?

- Data >>> Catalyst Fractals[18]

All possible data pattern decompositions.

Fractals are a means of seeing the embedded nature of information.

- Catalyst >>> Agent Network theory[19]

All possible relations between agents.

Network theory allows the abstract relations between nodes (that may be thought of as agents) to be analyzed.

- Agent >>> Catalyst Category theory[20]

All possible embeddings of agents within agents.

Category theory allows the onto and into mappings of embedded agents to be analyzed.

These methods perform a different kind of task than those minimal methods associated with design. They allow the connection of specific

[18] Reference?
[19] Reference?
[20] Reference?

distinctions to be analyzed as they relate to the field of all possible distinctions. From this perspective the Catalyst is the means by which a myriad of specific distinctions and decisions are made within the realm of the WHY of the software theory. There are a myriad of reasons why any piece of software is the way it is. This is seen as what Peter Naur[21] calls the non-representable nature of software theory. All we can really have is partial views which do not deal with WHY but instead with WHO WHAT WHEN & WHERE. But all these myriad reasons why are based on specific distinctions and decisions made somewhere in the history prior to or within the project. The Catalyst is the agent which motivates these decisions within each kind of work other than design. The design has specific decisions as well, but they do not deal with the relation of the theory to the world. The catalyst deals with the specific decisions in each kind of work that connects the software theory to the world. It is called the Catalyst because the viewpoint demands or inspires the requirements analyst, the implementor, the integrator or the tester to make these distinctions and decisions that connect the software to the world and make it

[21] Reference?

more than a theory. The catalyst does not make these decisions. It is a viewpoint that demands that these discriminations be made, not the subjectivity that makes them. The implementor, for instance, is driven by the viewpoint of the Catalyst to create a myriad of ones and zeros using a particular compiler in order to render the design concrete. The Integrator is driven by the viewpoint of the catalyst to put all the pieces of the software together and distinguish those that fit together with the hardware and eachother, and those that do not. The Tester is driven by the Catalyst to distinguish those tests that the software passes and those that it fails. The Requirements Analyst is driven by the Catalyst to distinguish the thresholds of acceptable performance and the limits of desired functionality. The software design itself is an abstraction. Without the work of the Catalyst that connects it to the "real" world it would remain a non-fully representable theory. When the representation is realized in required, implemented, integrated, tested code, then it becomes more than a theory. It achieves full representation in which the gap between the partially ordered and fully ordered viewpoints are jumped over and through the realization that the always flawed embodiment is made complete. Like Zeno’s paradox in which the arrow never reaches the target, without the Catalyst the non-representable theory of the software design would never be realized as a working system.

One way to view the Catalyst is as the positive face of the Essence of Manifestation[22], or intersubjective unconscious[23], that refuses manifestation at the meta-technical level of manifestation. The Catalyst is a cornucopia of distinctions; it is a source of variety which floods the software practitioner with all the myriad details that have to be just right which make the job of writing software so difficult. Where the essence of manifestation (which Deleuze and Guattari call the Body-without-organs[24]) is never seen but only its effects are seen as distortions, so the Catalyst for distinctions is seen everywhere. Gregory Bateson called these distinctions “differences that make a difference”[25]. A software system is just this plenum of minute distinctions which is best epitomized by the ultimate pattern of incomprehensible zeros and ones of the

[22] Michael Henry THE ESSENCE OF MANIFESTATION ?

[23] Carl Jung called this the ‘Collective Unconscious’.

[24] G. Deleuze & F. Guattari ANTI-OEDIPUS

[25] See STEPS TO AN ECOLOGY OF THE MIND

compiled code. The compiled code is the source of the behavior of the software system. Behavior is the manifestation of autonomy in action. The artifact that produces the behavior is opaque or incomprehensible to us. It is exactly the opposite of the transparency of functionality. This artifact of human endeavor which is incomprehensible actually operates when executed, doing many things which are comprehensible as it ties together the technological system and coordinates its actions. Thus, we can see these two as nihilistic opposites which cancel to produce the next higher meta-level of Wild Being, just as Process Being and Nothingness cancelled to produce the meta-technical level of Hyper-Being.

Since the autopoietic ring itself is a natural product of intersubjective cooperative work or discourse, we can see vestiges of it in the history of human civilization. For instance, it shows up in the Chinese archaic sciences as the ring of Five Hsing or transformations. In fact, where the West traditionally produces control-oriented models of systems, we can see now that Chinese archaic science may have been totally preoccupied with producing models of autonomous closed systems. So just as Western perspective lines converge while Chinese perspective lines diverge, so too here the Chinese views of science could perhaps have been diametrically opposed as well. The model of the Five Hsing is very ancient and is the basis of Chinese Acupuncture and Herbal medicine that the West is just now discovering in spite of its lack of Western scientific explanation. Perhaps that lack of explanation results from our not knowing how to describe autonomous closed systems. Now we can see that there are other sources of knowledge of how these autopoietic rings operate which are already available. In a recent book Varela[26] uses the concept from Buddhism of co-dependent arising to explain the relations between the links in the autonomous ring. Co-dependent arising says that these elements of the autopoietic network arise together and are mutually supporting with no outside cause. Each one causes all the others and ultimately itself through the feedback and feed-forward of the network itself. There is no cause outside the network. The origin of the network is always already lost. The elements of the network co-evolve over time arising from a

[26] Reference?

single origin with emergent properties which cannot be reduced to a lower level of structure outside its own self organization. In our case the five essential transformations of software arise from the hacking of the isolated programmer under the need to make that work intersubjectively accessible. This intersubjective alteration of hacking is differentiated under the same constraints that cause the Moods to arise in language, and each transformation takes on some characteristics similar to the moods. But once formed, because we are dealing with software, the essential processes become a self-generating ring which is autopoietic in nature. This means it is a closed unity which has exactly the same form as seen in the five transformations at the basis of the Chinese medical system.

This use of Buddhism to understand the closed autopoietic system suggests we might look at other Eastern sources as well. And the structure of the Five Hsing is an excellent example that fits right in to our study of the five essential software transformations. Strange as this may seem, we would well consider the relation between the production and control processes that the Chinese saw between the

Five Hsing.

PRODUCTION CYCLE

- Earth produces Metal
- Metal produces Water
- Water produces Wood
- Wood produces Fire
- Fire produces Earth

CONTROL CYCLE

- Earth controls Water
- Water controls Fire
- Fire controls Metal
- Metal controls Wood
- Wood controls Earth

Notice that true to autopoietic form these two cycles form a ring. Each of these operations has its inverse so there is both feed forward and feed back. The standard image for relating the different elements is a kettle standing on the Earth, over a bunch of wood logs, which are burning, heating the metal container that has water inside. The water boils and lifts the lid with its steam so that the escaping energy is called CHI. The Five Hsing work together to produce the flowing energy within the whole system that has an external effect. This energy has a pattern which is called the LI. These two concepts will be explored in further depth in the next section. Here our interest is on the Five Hsing themselves. They are each transformations that form a closed system. That system is seen as the basis of energy transformation in the body. It is what allows the human body to be an autonomous system. Thus, in the Chinese system it is this set of transformations that is directly responsible for the autonomous behavior of individual creatures.

Now it is interesting to connect the chinese transformations to those that occur in linguistics and software engineering.

- Earth -- Catalyst -- No Order -- Distinction
- Metal -- Agent -- Partial Order -- Autonomy
- Water -- Data -- Full Order -- Space
- Wood -- Function -- Partial Order -- Intentionality

- Fire -- Event -- Full Order -- Time

These five viewpoints form the vertices of a pentahedron in four dimensional space. The pentahedron has five points, ten lines, ten sides and five tetrahedral solids. The group of rotations of the pentahedron is called by the mathematical group theorists "A5" with 60 different group operations. It shares the same group of rotations with the Icosahedron. A pentahedron is actually two intertwined mobius strips. Since the pentahedron has five points or viewpoints and also five tetrahedrons, we can see that the tetrahedrons correspond to the five phases within which the transformations take place. So for this geometrical form the viewpoints and the phase structure is isomorphic. The structure generates by the combination of the viewpoints five phases which we have identified with the linguistic moods and the intersubjective software transformations. The pentahedron gives us a closed form in which we can go back and forth between viewpoints and minimal systems or phases.

As Buckminster Fuller noted in his masterwork

SYNERGTICS[27] it takes four independent co-appearing and overlapping phenomena to produce a minimal system. Thus, the permutation of the five viewpoints taken in sets of four produce the minimal systems of the separate phases. Fuller relates the minimal system to the tetrahedron alone. But actually, the minimal system has four faces that need to be considered which appear as TETRAHEDRON, KNOT, MOBIUS STRIP, & TORUS. Each of these geometrical forms contain 720 degrees of angular momentum. This number is a topologically significant threshold[28]. But these are merely geometrical figures that represent deep structural relations between aspects of the minimal system. The tetrahedron is the structure of the system. It can be taken instead as a lattice of the faceting of the elements that compose the minimal system. The Knot is the self-interference generated by the system. We have already seen this as the interference between the four independent information flows. The torus is the closed ring itself ,and the Mobius strip is the paradoxical nature of global versus local features of the system. In the autopoietic

[27] Reference?

[28] Reference?

configuration all of these faces of the minimal system become relevant to our analysis of the autopoietic system. The pentahedron is composed of five interpenetrating minimal systems. These are generated by the addition of one point and four lines to a normal tetrahedron. Thus, there is heavy redundancy in the reuse of lines and sides between these five tetrahedrons. In this situation the five tetrahedrons are manifestations of a single tetrahedron with a single viewpoint interchanged. This single tetrahedron has its four faces operating within the system of its five projections in each of its projections. So we see the system of projections take on the ring-like structure of the torus. We see within each segment of the ring an interference pattern between four information flows. We see that there is a paradoxical global continuousness to the whole ring while locally it is segmented which is like the global/local paradox of the mobius strip. All these features appear within the autopoietic unity which we can relate to our four dimensional geometry of thought which builds upon B. Fuller’s three dimensional geometry of thought. But this geometrical analogy is merely our way of viewing deep structures that arise naturally out of autopoietic unities as they manifest in spacetime.

The structure of the co-arising co-dependent autopoietic ring as reviewed here using geometric analogies is a static structure, whereas the ring itself is never static. It is, in fact, dynamic and has its own energy structures and pattern forming regimes which will be discussed in the next section. There is a tension set up between the functional and autonomous viewpoints which can be seen as defining the relation between what the Chinese call Chi (flowing energy) and Li (pattern). One way to view this relation between Chi & Li is by seeing that the continuous ring structure is the prerequisite for the production of Soliton waves which move within the ring. Soliton waves do not dissipate energy, and are basically entropically stable. Soliton waves can pass through each other and bounce off the sides of their channel without losing energy. The Chi may be seen as the movement of soliton waves around the ring. This ring itself is a dissipative structure that is negentropic but within the channel of the ring move non-entropic energy patterns as the Chinese have predicted. The Li would then appear as the pattern of standing waves that is set up in the whole ring structure by the flow of the soliton waves around it and their interaction with normal waves that would correspond to the input and output energies of the system. The tension between Chi & Li is adumbrated by the Catalyst which resides within the membrane of the autopoietic unity. The role of the Catalyst has been studied by Zelany. He has produced computer simulations of autopoietic systems which build their own membrane by the action of a catalyst on a flowing substrate that moves through the membrane. These studies show that autopoietic systems can be modeled as dissipative structures that are self maintained through the flow of an artificially maintained non-equilibrium. This is why the set of essential processes are called a WATERFALL model. It is recognized that it is a continuous flow through the stream of essential processes. What was not realized was that this waterfall, like those in Escher drawings, are self feeding and self organizing.

Another related speculation is that the connection between the five viewpoints is the source of language. Language is behavior coupled with meaning that is dependent on discrimination within the spacetime continuum.

This fusion of the Five Hsing into a single structure allows the human being to build up his world. Heidegger identifies three existenials: Discoveredness, Talk, and Understanding. Talk may be seen as the intersection of the Five Hsing within man. Talk is based on a whole series of levels of distinctions which range from phonemes, to words, to clauses, to sentences, to paragraphs, to discourses. As has been shown by Conway,[29] these distinctions are built in, and even the newborn baby resonates its whole body to these different levels of linguistic distinctions. These levels of distinctions are patterned by the syntax of the language which is for the most part a behavioral production process based on deep-rooted grammars such as those described by Chomsky in his transformational grammar. But all speech has a semantic dimension as well which corresponds to the functional or intentional aspect of the Hsing. Finally, the dance of language occurs in spacetime with the creation of patterns of sound. Thus, in language all the five viewpoints are represented as major constituents. And language is our major way to understand and even project our world. Thus, it

[29] Reference?

appears that the existential TALK is an intersection of our capabilities to produce everything else. Of course the other existenitals are Discoveredness which is the raw fact of finding ourselves present in the world *AND* the understanding of that world which is not limited to understanding through language. However, of these the prototype of the creation process itself is embedded in language, and we can see the Five Hsing clearly there in that arena. Those elements get changed when we move from spoken language to written language. The fact that we can understand both is some proof that understanding is not limited to language but is a separate existenial. In written language these same elements are present, though, in an altered form. Writing is based on a similar hierarchy of distinctions such as letter, word, clause, paragraph, text. Writing is a behavior which produces something with intended meaning. Writing encodes information in a spacetime process. So all the basic Hsing are interacting here similar to the way they interact in speech. However, there are some major differences. First the distinctions are in something that leaves traces, such as letters on parchment. Second the behavior that produces the communication is separated from the reading process in a way that listening is not separated from speech. The meaning is dependent on a reduced context of the text itself and is not situational. The spacetime process of writing has to do with manipulations by the hands instead of by the mouth and larynx on a different medium. The new medium makes possible the saving of the work for future reference which is the basis of our culture. But also all the effects peculiar to text pointed out by Derrida[30] come into play as well.

M. Zeleny and N. A.Pierre in their article*"Simulation of Self-Renewing Systems"* [31] enumerates the principles of management of human systems that he draws from his computer models. It might be well to review these in preparation for attempting to return to the problem of workflow moving through the autopoietic cycle.

- "Complex and dynamical human systems are to be *managed* rather than be analyzed or designed. Human systems management is not systems analysis or design."

[30] OF GRAMMATOLOGY ?

[31] In Erich Jantsch and Conrad H. Waddington (eds.), Evolution and Consciousness: Human Systems in Transition (Addision Wesley 1976.

Although we talk about applying the essential processes of requirements, design, implementation, integration, and test to the development of a process for doing those same things to develop software, we do not mean that one person designs the process and another person executes it. In this case there is a split in the autopoietic unity, and the whole thing falls apart into an allopoietic (other produced) system. At that point the law of requisite variety (R. Ashby[32] and S. Beer[33]) comes into effect. That law says that a control channel must be as complex as the thing controlled to exert complete control. Software processes are so complex that no communication channel that complex is practical. Only if the people enacting the process are the designers of the process, can they have any hope of controlling it. All external control is in fact illusory. Thus when Zeleny says that human systems are managed, not designed, it is clear that he means that management does not design the autopoietic processes but merely seeks to influence them indirectly by acting as a Catalyst through a narrow communications channel. This breaks the illusion of complete

[32] Reference?

[33] Reference?

control and makes management a resource (instead of some grand designer divorced from the work being done) used by the enactors of their self-defined process.

- "Human systems management is a process of *catalytic reinforcement* of a dynamic self-organization and bonding of human components. It does not design a managerial hierarchy of command and control."

Managers are catalysts which attempt to get an autopoietic sociotechnical formation instituted and then stand back to act as a resource for that once it is in operation. But as we have seen, management is not the only catalyst. In fact, the catalytic perspective is important in each of the other phases outside of design. The point is that there needs to be many catalytic reactants in order to keep the system in non-equilibrium. This is the main functional role of the catalyst. So in requirements we see that the requirements are always changing. This is a natural formation that occurs. In fact, it appears in all the non-design phases. In Implementation there are always changes in the underlying development system or the target system; in Integration there are always changes in the set of builds and the way the system is linked and configured; in Test there are always undiscovered bugs in the system to be routed out. All these points of continual production of variety must be managed. But they are essentially the generators of non-equilibrium that the autopoietic unity feeds off to keep itself going. We continually talk of getting rid of these nuisances. But, in fact, it is these producers of variety that are what keep the autopoietic system going. The autopoietic system is set in motion by the catalytic action of management. But the unity itself is a series of catalytic components which form a minimal system and keep the unity in existence as a waterfall of produced variety which comes about as a reaction to the catalytic viewpoint.

- “Components of human systems are *humans*. As such, they differ significantly from other components, mechanistic or biological, in their ability to anticipate the future, to formulate their objectives, to plan for their attainment, and to make decisions. These properties are sufficient to make

> human systems quite distinct from all other systems.”

It is important in our understanding of software process to keep this point in mind. We are designing human processes. We need to design them in such a way that they take advantage of human creativity and the ability to distinguish the correct way of doing things. Many process designers seem to forget this and treat the practitioners on the same level as computer hardware, seeing process only from control perspective. In the development of personal process and the recognition that human autopoietic systems must have enactor-process designers, we have attempted to build in this important precept.

- “The integral complexity of human systems can be lost in the process of its simplifying reinterpretation by the rigor of mathematical mechanics. Human systems can be described and studied through a relatively simple set of linguistic, fuzzy, and semantic rules, governing the self-creation of its complex organization. Human systems

> management is not operations research, econometrics, or applied mathematics.”

Zelney sees process from the point of view of cellular automata. Each actor uses a set of rules to produce his own variations on the process model. In fact, we see each phase of an autopoietic unity as such a cell. This is closer to Varela’s cellular automata models of autopoietic rings. Thus, it is the phase of the autopoietic unity that is the “cell” not the human being. Given that change in emphasis, it is clear that the autopoietic ring can generate much more variety than any control-oriented structure. The human that is enacting a particular kind of work executes the rules associated with a particular process. The fact that these processes should be described in terms of fuzzy or linguistic variables is fully supported. In fact, we see an interference phenomena at the center of each phase in the ring where internal and external information flows come together. This knot of interference needs to be described using fuzzy techniques because the complexity of the interference pattern itself is far beyond what any human being can handle. Thus, a fuzzy interpretation occurs in every phase of the ring by which the interfering information flows are reduced to some common denominator which the human can handle.

- “Interactions between components are not those of electronic circuitry, communication channels, or feedback loop mechanisms. Rather, they are organic and dynamical manifestations of organizational autopoiesis. Human systems management is not cybernetics or the information theory of communications.”

This point is just a statement that autopoietic unities are emergent phenomena. They cannot be reduced to other phenomena without doing violence to the autopoietic unity itself. In fact, if we reduce them, they become invisible. This is why they are not seen by control-oriented Western science. Unless you are looking for closed unities, they do not appear. In fact, the two viewpoints are probably mutually exclusive. The Chinese did not see control-based systems because they saw everything as closed unity. This goes along with the fact that their worldview did not have transcendentals but was entirely immanent from our perspective. It is our dualistic tradition which sees every relationship as a power relation which cannot “see” closed unities. In the non-dualistic Chinese tradition complementary opposites are seen instead of power relations. Probably there are many closed unities right in front of our very eyes if we could but see them. An example that springs to mind is the family. The family is a self-differentiating unity which is having trouble surviving in our present society. But internal to the different types of family there is probably another different set of autopoietic phases. It is not the members that are the autopoietic elements, but different sets of behaviors which the members process in sequence. To recognize autopoietic unities we must shift from looking at the “entities” to the processes, and then we must look for circular intersubjective behavioral processes. In this light it is important to recognize the special place Symbolic Interactionist[34] social theories have in being in tune with the autopoietic viewpoint on sociotechnical systems. Symbolic Interactionsism focuses on behavior and thus treats autonomy without the reductionism of pure behaviorism. For Symbolic Interactionsim it is exactly the self that becomes the center of

[34] See Patrick Baert and Jan De Schampheleire “Autopoiesis, Self-organization and Symbolic Interactionism: Some Convergences” in Kybernetes 17, 1 p60-69

attention when considering what controls behavior. In autopoietic systems the nature of the self becomes the crucial question. The self for G.H. Mead, the founder of Symbolic Interactionism, is an internal reflection of how we think others see us. It is this reflection that controls behavior of the individual. We would expect in every social context where some group organizes itself that these autopoietic unities may arise. But not all human social groups are autopoietic unities. Autopoietic systems are a special type of social formation which is perhaps rare. A textbook example of an autopoietic system is given by Plato in his LAWS. Despite the criticism that he has received for the totalitarian aspect of his "closed society," it is clear that he had the ideal of creating autopoietic social unities in mind.

- "Dynamical order of human systems organizations is maintained through a continuous renewal of certain *nonequlibrium conditions*. Both nonequlibrium and instability are essential for self-organization or higher complexity. Human systems management is not a theory of general equilibrium."

Autopoietic systems are Dissipative Structures in the sense defined by Jantsch[35]. A dissipative structure produces a membrane which represents an negentropic gradient which creates organization despite overall entropic dissipation. The autopoietic system is exactly the same thing. It produces an organization which is negentropic in spite of entropy at the component structural level which is constantly being replenished in order to fight the continuing always already lost battle against entropy. Thus, we do not think of autopoietic systems as things. They are by definition processes only, and as soon as the process stops, they cease to exist. Entropy disperses their structural components. An excellent example is the cult which has become such a popular religious form in American society. These closed social systems with their own behavior patterns and beliefs many times greatly at odds with predominate values flourish in the overall materialistically-oriented dominate society. The membrane of the cult itself must be continuously maintained, and once the members stop maintaining their difference from the social environment, the cult vanishes. All autopoietic systems are a little

[35] See Jantsch, Erich (1975). Design for Evolution: Self-Organization and Planning in the Life of Juman Systems: New York: Braziller.

like cults in this sense of being a closed intersubjective system. In software we call them the team. They develop their own sets of values, specialized vocabulary, group memory, etc. which mark them off from everyone else. Everyone knows who is in the team and who is outside. The boundary of the team is continuously being maintained and augmented as a spin-off of all the activities of the team, such as meetings that include and exclude certain people within the overall organization.

- “The concepts of optimization and optimal control are not meaningful in a general theory of human systems. Human aspirations and objectives are dynamical, multiple, and in conflict, as are those of human organizations. This conflict is the very source of their creative evolutionary unfolding. Human systems management is not optimal control theory or theory of conflict resolution.”

Human beings are variety producers, as Stanford Beer has said. Once the variety is produced, then it becomes the field for interpersonal conflict. But the focus on conflict is probably a throwback to the old control-oriented way of viewing human systems management. Rather, it is important to take the view exposed by Deleuze and Guattari in their books on Capitalism and Schizophrenia that are a study in the organization Wild Being, which sees human society as at root schizophrenic. This realization that the process of variety production is endemic and will never go away is very important. From the old control-oriented viewpoint this looks like the natural appearance of conflict at the root of all social structures. In the LAWS Plato calls this the war of the ALL AGAINST THE ALL. Plato goes on to show that the laws must seek to instill all the virtues, not just courage, but also justice, moderation, and wisdom. Thus, we must learn to accept that human beings will produce variety, and we cannot ever control that effectively with any process we might invent. Therefore, we must seek to only reduce unnecessary variety and do that by getting the practitioner to define his own process via a set of high level guidelines. In extremes the variety production goes out of control and is seen as a sickness like schizophrenia. The ad hoc initial maturity level software processes suffer from this syndrome. All we are doing is attempting to channel the variety to a certain degree. If we can get an autopoietic system set up, then we expect it to organize itself creating the requisite variety for its own problems but reducing the wasted effort of creating unnecessary variety which in extremes becomes schizophrenic. It takes more energy to create new variety instead of reusing and following old trails. In fact, in process work we are attempting to produce what Waddington calls “cheords”[36] which are paths of least resistance to serve as the river bed of the autopoietic unity. The autopoietic social unity organizes itself in a channel of least resistance which is laid down for it to follow. If it needs to deviate, then it decides to spend the resources to produce the necessary variations and perhaps develop a new channel. Other than that the autopoietic unity would be expected to reign in its own variety production and as part of its own self-organization produce only the requisite variety of the task at hand.

- “The inquiry of human systems is *transdisciplinary* by necessity. Human systems encompass the whole hierarchy of natural systems: physical,

[36] See TOOLS FOR THOUGHT ?

> biological, social, and spiritual. Human systems management is not interdisciplinary or multidisciplinary; it does not attempt to unify scientific disciplines, but transcends them.”

Those engaged in the development of software processes have a lot to learn from other disciplines in the humanities. So the interdisciplinary and multi-disciplinary aspect of software process definition will increase in the future under the breakdown of the purely technical solutions being proposed now. However, what is not well recognized is the transdisciplinary nature of autopoietic unities. They are ***trans***-disciplinary because they are rooted in ontological structures that determine how manifestation occurs in our worldview. They are rooted in the structures of Being which has fragmented into a series of meta-levels with new entities entering our world founded on specific kinds of Being. Software is one of these new entities. Artificial intelligence techniques is another source of new entities from an even higher meta-level of Being from that of software. Whenever we see specific structures such as the autopoietic software development ring that are representative of the structuring of our worldview itself, then we have entered into the transdiscipline arena. All disciplines that recognize and study autopoietic systems will eventually cross pollinate as those that study chaotic systems now do. But it is yet another step to recognize that we are acting out structures that articulate the general structure of our woldview and that our exploration of human sociotechnical systems has implications for our development of philosophies underlying our work structures.

8. THE WORK FLOW AND PROCESS WISDOM

With the definition of personal process and the recognition of the nature of autopoietic systems we are in a unique position to differentiate the meta-levels of process refinement. The foundation of all process work is the generic process definition. There are a myriad of ways to define generic organizational processes. Some researchersl want to treat people as machines and program them like computers, saying that process may be captured in a formalism like a programming language. Other industry people implementing process in their own organization think we need to provide desk instructions that tell their people every move to make. There are others that merely want something that passes inspection by SEI assessors as a workable process definition. In our own case we are concerned with actually improving our work, and so we see process descriptions as an uneasy marriage between a description of what already goes on and a description of our best practices with a sprinkling of mandated improvements. However, all proscriptive process improvements assume a continuity of underlying work that can be ordered according to process guidelines. The envelope of continuity is different for different teams, individuals and projects. But each envelope is construed as an ongoing context within the work situation which can be entered and re-entered in midstream to keep some aspect of the work to be accomplished going. This continuity is socially constructed. It is a product of negotiations for resources and agreements as to what work needs to be done to accomplish the final ends of the project. It is the woof that runs through the warp of the schedule of the project. But in some way the proscriptive process is assuming a worker (subject) who is continuously putting effort into that context maintaining the forward momentum of that aspect of the project. This forward momentum is something of an illusion as is the abstract description of processes that assume ideal workers under ideal conditions. This is, of course, the level at which SEI sanctioned process definition occurs. We can see that this level of proscriptive process corresponds to the ontological level of manifestation represented by Pure Presence[37].

[37] See M. Heidegger BEING & TIME

In execution projects always turn out differently than they were planned. Continuous effort is dependent on a myriad of constraints holding steady, and other activities going well and producing the necessary outputs on which immediate work is based. These constraints are continuously being trespassed so that the plans are always breaking down and having to be reformulated. This adaptation of the project in midstream is necessary to maintain the illusion of continuity. In reality the continuity is constantly changing and adapting as the project is moving forward. This level of adaptation is where the negative form of breakdown is aperiodically occurring as the plans attempted to be turned into actualities. Probabilities reign here as in every case where determinate plans are actualized. This is the level at which statistical process measurements are taken and used as a basis for global optimization of the overall process on particular actual projects. We can see that this level of statistically enacted process corresponds to the ontological level of manifestation represented by Process Being[38].

The next deeper layer of process is that which

[38] See Heidegger BEING & TIME

personal process addresses. It addresses the positive aspect of breakdown where all work is redefined on the basis of someone’s understanding it in a new way. Personal process is directed at getting a picture of the work from one’s own perspective in order to understand it well enough to redefine it. Whereas the top layer assumes work has a long lasting form, and the second layer is just making local fixes to get back on track, this third layer is actually actively trying to transform the work into a new pattern by understanding it thoroughly from a particular perspective. Every kind of work is susceptible to redefinition. This is the arena in which all real process improvement must take place. It is the exact arena ignored by SEI. The statistical analysis performed by process improvement specialists see individuals as random variations. However, it is individual initiative and insight which can really make the big difference in redefining process creatively to streamline it. And this is also the level where our culture has its greatest strengths in providing individuals who are able to take initiative and make changes that make a difference in the way things are done. It is ironic that American business culture has emphasized monolithic organizational structures and multiple tiers of control that have mitigated against the very forces of innovation spawned by our culture. In many instances creative individuals have either had to start their own companies, go elsewhere to work where they were given the power to make changes, or keep quiet and let things remain bad. We can see that this level of personal innovative process corresponds to the ontological level of manifestation represented by what Merleau-Ponty calls Hyper-Being[39].

New team organizations offer the potential of supporting individual initiative. The layers of stultifying management have disappeared. However, the problem in the new organization is that if your idea for improving things is beyond the scope of your project, then there are no resources to draw upon to make it happen and no one with broader concerns with the power to make it happen. Yet the resource that the team organization has to offer is the close knit group with a common goal so that if your idea for improvement falls within their scope, then it is more likely to get implemented if they can be convinced of its efficacy.

[39] See THE VISIBLE AND THE INVISIBLE; Heidegger calls this ~~Being~~ (crossed out).

The final and deepest layer of process is that of the line of work and going concern. This is the layer which addresses what actually happens in all its gory detail and irrationality. It is the land of political decisions, wrong goals, personal conflicts, and everything that is counter productive. It is also the layer in which the person’s own motives and behavior patterns exist; where social alliances and personal networks have real effects on the way things actually get done. It is many times referred to as under the rubric of the Jungle. Deleuze and Guattari call this the schizophrenic level of society[40]. At this level there are aspects of personal and social behavior that will never be rationalized and that should never be attempted. It is the level at which personal freedom of action reserves the right to reign unfettered. Yet this is also a level under attack by automated systems which can track personal productivity keystroke by keystroke in automated routinized jobs. SEI only implicitly recognize this level in their classification of initial maturity as ad hoc processes. We can see that this ground beneath all processes corresponds to what Merleau-Ponty calls Wild Being[41].

[40] See ANTI-OEDIPUS

You will notice that the SEI model completely ignores the third layer at which personal process appears. Their model is not culturally adapted. It does not take into account the creativeness of the individual that has the highest potential impact on changing the process for the better. It, in fact, assumes the old control structure oriented organizations that have a major incentive to stifle creativity. Culturally this tension between control orientation and the natural percolation of innovation from self-directed individuals represents an untapped resource for process improvement. The connection of this resource with the team-centered environment is potentially a means of deriving great benefits in efficiency and quality improvements. By giving the practitioners control over their own processes and encouraging adoption of creative solutions, we can anticipate great gains. But this mitigates against legislated proscriptive processes imposed from above in the old control oriented style.

Processes are streams of predefined actions called tasks, streams of information such as communications, streams of thought and self-

[41] See THE VISIBLE AND THE INVISIBLE

reflection, streams of materials that are worked upon, streams of representations that are transformed. As streams they represent the combined energy of groups of individuals working in concert. As such they appear as a dimension of our work lives that have been hidden by the static control structures that attempt to govern these streams. Organizational structures appear as static. Facilities structures appear as static. People who do the work are considered as replaceable roles. The old control structure has taken the assembly line as its model with not only replaceable parts, but replaceable workers with limited specialized skills. In a shrinking workforce the emphasis is shifting to multi-skilled experts and cross training. The functional organization is slowly disappearing. With this shift the previously invisible realm of process streams are becoming visible. Teams own their processes. The work is shared, and the complete product is a visible measure of team performance. The workers are multi-skilled and enmeshed in a compact sociotechnical system where workflow is more important than concrete reified products because it has finally been realized that the product is only as good as the process used to develop it.

When we enter into the realm of process, we are really in an arena in which we do not know how to deal with things. We are so used to concentrating on the final result that it is difficult to even see the flow of incremental actions and thoughts that results in the concrete product. Here we need a different non-object centered vocabulary such as that developed by the process philosopher Whitehead, or some other that allows us to deal with streams rather than the reified in products of streams. When we look around for such a viewpoint that will allow us to understand the streams of work and describe them directly, the model that first comes to mind is that developed by the Chinese. Slowly we are realizing that the Western worldview is inadequate for dealing with evolutionary and continuously changing fluctuations in the world. So we are led to look further and further afield for a way to get a handle on these phenomena. The Chinese had in their civilization a whole way of looking at the world that dealt with it in terms of fluctuations instead of reified objects. Here we will introduce a few concepts from archaic Chinese science to show how they apply to our level of personal process articulation.

With every consideration of process the basic question is how to differentiate sub-processes within the overall flux of life. That flux of life will be called here primary process, and the question is how we distinguish other secondary processes that are embedded in primary process. The Chinese had three concepts that they applied to this problem, and we might well learn something from them in our own exploration of process. The three concepts are SHU (Countability), LI (Principle), and CHI (Flowing Energy). At the level of primary process or the flux of life these three concepts are fused and indistinguishable. The secondary autonomous processes cannot be counted or distinguished, and Principle is the same as the Flowing Energy of the process. The secondary processes appear when a differentiation occurs under the action of the Catalyst. In our case it is the differentiation of kinds of work. This differentiation is subjective and tied to the observer. Each observer can see a different set of process classes in operation depending on his viewpoint. However, once this differentiation occurs and different kinds of workflows are distinguished by an observer, then LI (Principle) becomes distinguished from CHI (Flowing Energy). Note here that it is not a matter of distinguishing process from products. Products are not as crucial in this way of looking at things as it has been in our Western worldview that continuously attempts to reify processes into things. Instead the major differentiation is between the flows of energy and their patterning. The patterning, according to the Chinese, always manifests a principle called LI. In a view of things that sees process as manifestation or presencing, it is the inner coherence of the interweaving processes that is important, not the reification of streams into things.

Let's take the example of the growth of a tree. A tree grows by manufacturing energy and using it to build up the structure of the tree at a cellular level that hardens into the dead core of the tree. Only the outermost layer of the tree is alive. This process of energy production and transformation into structure using photosynthesis and materials taken from the soil is an ongoing process. But as the process goes, it lays down the rings of the tree layer by layer in a certain specific pattern. This patterning specific to each tree, to each species of tree, to trees in general displays the principle that the tree manifests. Seeing this pattern is of the utmost importance in the Chinese way of looking at things. One is looking not at the products per se, but instead at the total pattern of the interplay of sub-processes as they lay down a pattern in their products. One is focusing on the traces of the intertwining streams of activities, not at individual products as entities separate from each other. The LI or principle is implicit in the patterning of the relation of secondary processes to each other. It is observed in the traces left by these streams in the total pattern of the products. The LI can only be separated out from the Energy Flow of the secondary processes by looking at their interrelationships. The LI can also be inferred by the patterning of all the secondary processes products. If you cannot see secondary processes as countable and separate, then the CHI and LI are indistinguishable. CHI is the energy flows themselves. These energy flows each have their own quality that can be distinguished in the envelope of countability or distinguishability. The quality of the process is something we intuit directly. It is completely subjective even if different people can distinguish the same processes. But it is a very important element of the analysis of processes that we are not used to dealing with. Normally if something is not objective, it is dismissed. However, for process improvement what feels right or looks right is very important. Quality of process is an intangible, and we need to have a way of talking about the differences between these intangibles if we are going to improve process. The idea that metrics is the only way to improve process is completely wrong. It takes into account quantity but not quality. We need a balanced approach to process improvement that equally emphasizes quality and quantity. CHI is merely the evolution or fluctuations of energy that has different qualities within the area demarked as the providence of a particular secondary autonomous process.

Finally we should mention tertiary processes. A tertiary process is embedded in a secondary process in a manner similar to the way secondary is embedded in the primary process. The tertiary processes are important for all technical work since they represent the spinoffs set in motion by the secondary process. For instance, in design process one might start some automated consistency checking tool working on a design representation. The process of consistency checking is done independently by the computer but set in motion and guided by the secondary process. If the secondary process does not set it in motion, guide it and integrate its results, then it merely dies without being able to contribute anything to the overall process. Tertiary processes may be represented by other people. When you get someone to give you a partial solution to help you in your work, then you have set off a tertiary process that must be guided by you and the results of which must be re-assimilated. To that other person though, what you considered a tertiary process may be a secondary process because to them that is their main line of work. Tertiary processes, like secondary processes, are countable, have flowing energy and exhibit the inner coherence of principle, both in the patterning of the interaction of streams and in the patterning of the products. The major difference from the point of view of the observer is whether this is the major ongoing kind of work being done or whether these are spinoffs that represent invocations of other processors.

The Chinese ideas about process are very sophisticated, and this is just a brief introduction. It helps us to see process as a different way of looking at things. We could also use process-based philosophy to make similar points. But the Chinese ideas are simple and practical, whereas within the Western worldview starting from the position of assumed reification (NOUN CENTERED) we need some work to get to a non-reified process view of things (VERB CENTERED). But there is a considerable wisdom in the Chinese view of process. It tells us that we should be looking for the inner coherence of streams of work and not at the products. It says that each stream of work will have its own principle which we are approximating when we isolate it and its own quality of flowing energy. We can expect each kind of work to be experienced quite differently by the practitioners, and that is how they know one from the others. In kinds of work there are no clear cut boundaries. The boundaries are imaginary. But these imaginary boundaries are reinforced by the countable difference between the differentiated kinds of work and the qualitative differences between the streams as well as the different principles being manifested in the patterning of the interaction of the work streams.

Much work needs to be done to come to a full appreciation of the Chinese concepts and their application within our study of processes. Some cultural translation is definitely in order. But the fact that other cultures have succeeded in looking at the world in a non-reified way is encouraging. Our efforts must be toward attempting to find non-reified ways to describe and consider the enaction of processes. This is a difficult task. But it is central to the improvement of process centered working environments. As long as we look at processes from a product orientation, we will be stuck with terms and concepts inapplicable to our process centered way of looking at things. Process centered ways of seeing work is a paradigm shift. It calls for a major reformulation of the way we conceive and do work. If we cannot make this shift, then we will merely trip over our own shoelaces as we apply the old way of doing things to the new tasks before us. This transition to the process centered paradigm will take time and will not be painless, but it signals a major reformulation in the way we construct our sociotechnical system and the working world.

9. THE PROCESS FUTURE

Process is actually a combination of behavior and functionality within the sociotechnical system. Functionality is really intentionality. All the basic views which apply to software systems may be applied to the work of constructing software. These views are AGENT, FUNCTIONALITY, EVENT, DATA. The traditional life-cycle view of process is event and data centered. Events are milestones, and data is products. The new way of looking at process emphasizes teams of multiple agents and functional (intentional) goals. Kinds of work are merely the functions necessary to produce a product within a certain timeframe. The process paradigm shift now emphasizes teams and work functions rather than spacetime ordering that the old control centered organizations emphasized. To them the process of getting to milestones and producing products was a black box. Now we are emphasizing close cooperation of a team which takes responsibility for all the kinds of work necessary to produce a product using an adaptable spiral lifecycle.

It is important to realize that the agent and functional perspectives are very different. In Maturna’s description of autopoietic systems he clearly distinguishes between organization and structure and between the closed and open aspects of living systems. In his work it is clear that the autonomous individual that expresses self organization (autopoiesis) must be considered as a closed organization; that the openness of systems where that individual reacts to inputs and responds with outputs is only loosely coupled to the closed system of the individual who is organizing itself and maintaining its organization. In such an individual the structural components might change, but the interrelation between those components is maintained. Here an example from work organization might be the relation between work assignments that are different for each individual and the pattern he maintains between his processes for dealing with those work assignments. Clearly the person who is governing his behavior in order to maintain a specific disequilibrium that allows him to continually respond to new work assignments is self organizing. That disequilibrium is the pattern of his personal work processes. The maintenance of the disequilibrium between functions is related to the autonomy of the individual. The individual may be seen as the carrier of this pattern of disequilibrium. The functions themselves are the channels of response to the outside world. The maintenance of the disequilibrium is a closed hysteresis loop which continually attempts to maintain itself or seek a new stable disequilibrium point at which a new pattern of processes would appear.

Thus, there is an inherent tension between the autonomy of the individual as a closed system and the openness and response to the environment. This tension must be addressed by any theory that attempts to show how process works in the sociotechnical system. The tension between open and closed, between function and autonomy, between process and team structure, defines the interface across which SHU (countableness), LI (principle), and CHI (qualitative flowing energy) show up. We are constantly giving different kinds of work to different people based on skills, and the same kind of work to different people who perform differently. This constant redefinition of who does what and how well is exactly where we see primary process being differentiated from secondary process. When the work is divided and assigned, then the different energies and qualities of the people interact with the different types of work to be done. This generates the differentiations of the qualitative flowing energy of the people doing the different kinds of work in concert. It also manifests in their interaction the combined LI of the people and the LI of the work. If you reassigned the same work to the different people of the same group, you would get a different result.

The basic principle announced by Lo Ch’in-shun in his book Knowledge Painfully Acquired (K’un-chih chi)[42] is that “Principle is one; its particularizations are diverse”. This means that the LI of the people and the LI of the work to be done is ultimately one from the view of primary process. However, as particulars are produced in the stream of process by the agency of specific individuals, diversity is produced which is like the shattering of glass. So give the same work to different people of the same team, and you will get different results. The manifestation of the LI of the people in the team and the team itself and the organization in which the team is

[42] Reference?

embedded is unified. In other words, the manifestation of fused LI and CHI in the whole of the organization is unified by the nature of manifestation. So any one thing you look at mirrors the whole. It is only when we differentiate from our own perspective that we see different secondary processes producing myriad diverse particularizations. The work of secondary process is to attempt to give the individual control over his portion of the process he is embedded in and give him a way to resonate with the rest of the team and organization so that the pattern of the LI is clearer, and the CHI or qualitative energy flows faster and clearer.

The differentiation of LI, CHI, and SHU occurs in the tension between autonomy and intentionality (functionality). The Chinese developed this process view of their world centuries ago. We are just now learning how to look at the world in a similar light. In the future of process development we have a long way to go as ours is a young civilization. But the way is cleared for us to dig deep into the process-oriented way of looking at the world and learn something about our world from the way we envisage our work. This says that the process view of work is the start of a fundamentally different way of conceiving work which will ultimately effect the whole economy and transform our whole culture. Process centered philosophies such as that of Whitehead are not that old, but already we are considering our work differently. Eventually we will realize that we are our work and that our products are not separate from ourselves. And the quality of the products we produce will become an issue of the quality of our lives.

This transformation of our concern for the quality of our products into a concern with the quality of ourselves is an essential change. A change that must begin with the overhaul of the educational system, and end with the revamping of the corporate structures. The changes that are needed to keep America competitive are too numerous to detail here. But the major point is that one of the results of any planned restructuring is the integration and harmonization of the various levels of process identified in this essay. Chang[43] has identified several levels of harmony:

[43] Reference?

- Logical Consistency Proscriptive Process
- Interaction Breakdown Adaptive Spirals
- Mutual Support Personal Process & Autopoietic Rings
- Interpenetration Self identification with own process.

The definition of proscriptive processes supplies us only the lowest level of harmony in terms of the logical consistency of the processes with eachother and internally. As we apply these processes to the work we do, we see that they interact in complex ways which are many times unpredictable. Thus, adaptive processes are necessary that take into account breakdowns and necessary changes in direction. The Spiral Lifecycle model of Barry Boehm is such an adaptive model for software development. But unless the processes become not only adaptive but also mutually supportive, then there is no real robustness within the processes as they are enacted. I think personal process comes in here as a means of creating dynamic interdependencies between the work which different team members are carrying out asynchronously but in harmony with each other. If each member is looking to redefine the work to make it more efficient, higher quality, with greater impact on the achievement of mutual goals, then through mutual negotiation a work structure will coevolve into something unexpected by everyone. Work which is laid out in advance as immutable is a roadblock to coevolution. Coevolution assumes ongoing real-time co-adaptation by people redefining the work together.

Redefining work is in someway a redefinition of ourselves. Thus, the motivational layer which appears in the lines of work and going concern ultimately needs to be integrated into our view of processes. They are OUR processes, not something imposed from without, but something we are creating ourselves as the means of defining our world. We are the world we define. So it is necessary to align our work processes with our own goals. This is difficult to do as long as they are imposed on us. But many companies are discovering that encouraging internal entrepreneurs allows them to maximize the relation between their employees, goals and the company goals. Employees are seen as owners, not as an external resource. Employees assume ownership and the rewards and risks are assumed by empowered individuals. This fosters an in-depth commitment beyond what is normally possible to achieve in control-oriented companies which view their employees as expendable resources.

The harmonization of all the levels of process is the ultimate goal of process. Process can be optimized as an external or objective feature of the organization only to a certain extent. In order to achieve the greatest possible optimization, the view of process as an external or objective thing must be replaced with the view that process goes to the very heart of who we are and that our work is the creative expression of who we are as individuals, teams, and organizations. This means changing how we see ourselves and each other. The utilitarian approach to others now dominant in our corporate culture can only reach the interaction level of harmony. In order to go beyond that we must be able to become mutually supportive and finally see each other as completely implicated in what each of us does separately. Interpenetration means that what I do is not just supporting of what you do, but that it is recognized as totally intertwined. At that point the line of work and going concern itself becomes the process which is completely enmeshed, but not as a hidden undergrowth. Rather the going concern and line of work is the organization and the individuals creating themselves anew over and over again as their work adapts and is completely transformed by redefinition. When we think in terms of continual redefinition of work as a result of personal process, it is clear that we are ourselves a product of the work we define for ourselves. Thus, redefinition of our work gives us the power over our situation we normally lack in the workplace. Empowerment of individuals to control their own processes makes them able to express themselves through their work. They are more creative, and are thus more fulfilled. There is less loss of productivity because of lack of motivation.

The future of process is a fundamental transformation in how we view ourselves and work. It is our challenge to navigate that transformation and arrive at a process that is harmonized. Process harmony requires a depth which is uncommon. However, unless we attempt to attain that goal, we will never succeed, and our efforts in the process arena will remain hollow and superficial. The development of process wisdom is the ultimate result of process harmony. Process wisdom comes from the continual attempt to attain process harmony. It is the kind of wisdom that the Chinese were able to develop before us. The age of their civilization attests to the value of that wisdom. The fact that we were able to destroy that civilization does not mean we Westerners are capable of producing a long living civilization ourselves. The process initiative is the first step toward understanding the ecosystem of the sociotechnical system. Whether that ecological view will allow sociotechnical systems to survive is a question still to be answered. But the future looks bleak unless it is possible to develop some process wisdom along the way.

So many people when they hear about process say, “We talked about all this before and nothing has ever changed.” They say, “We are still inventing our processes anew with every project.” They also say, “We never have enough money to do it right but always enough to do it over and over.” Software process improvement needs to address these comments directly. It can only be done by just saying, “You own your process.” No one can fix someone else’s process, except superficially. Invent your process anew if it saves money, or time, or improves quality. But do not reinvent when it costs more for no benefit. Then go beyond that to say, “You ARE your process.” This is the only way that real progress will be made in the trade-off between quality, and productivity. Optimization from outside is only a half measure. Optimization from inside is the key to real efficiency.

10. ABSTRACT

This paper begins with current AM&ASD division initiatives and goes on to explore the central issues in software process development with a view to uncovering possible future solutions.

11.KEYWORDS

Key words: Software Engineering, Software Process, Sociology of Technology, Autopoiesis, Software Methodology, Sociotechnical Systems, Software Assessment

12. BIBLIOGRAPHY

The following is a listing taken from one of Randall Whitaker's bibliographic database.

AUTOPOIESIS AND RELATED LITERATURE

by Randall Whitaker

The central subject is the theory of "autopoiesis" developed by the Chilean biologists Humberto Maturana and Francisco Varela. The listing is offered "as is" -- with occasional notes and whatever disjunctions were introduced by the clash between UNIX and non-English characters ;-} .

DISCLAIMER: This listing is being offered as a resource to others who have expressed an interest in autopoietic theory (and in the hopes of generating more such interest....). I am not soliciting (nor do I expect to honor...) requests for hard copies of the listed documents.

Comments and discussion are welcome.

Randall Whitaker, Institutionen for

Informationsbehandling / ADB Umea University

901 87 Umea SWEDEN
rwhit@cs.umu.se

LISTING DATE: 28 April 1992

****

Anderton, R. H.
Whither Cybernetics: Past Achievements and Future Prospects
Kybernetes, Vol. 9 (1980), pp. 289 - 293.

Andrew, Alex M.
Autopoiesis and Self-Organization
Journal of Cybernetics, Vol. 9, no. 4 (Oct.-Dec. 1979), pp. 359- 367.
Apparently the same text as Andrew article of the same title (1980). Claims that autopoiesis alone cannot suffice for studying self-organizing systems. Andrew states there are 2 additional features of SOS which much be accounted for: (1) the progressively succinct representations applied as “meta- representations”, and (2) the replication of existing “successful” structural features. Note that both entail some measure of teleology / purpose ascription and are therefore to be considered only from the perspective of a given observer.

Andrew, Alex M.
Autopoiesis and Self-Organization
Progress in Cybernetics and Systems Research, Vol. 7, Washington DC / London: Hemisphere, 1980, pp. 281-284.
Apparently the same text as Andrew article of the same title (1979).

Andrew, Alex M.
Autopoiesis-Allopoiesis Interplay
in Zeleny, M. (ed.), Autopoiesis: A Theory of Living
Organization, New York: North Holland, 1981.(Vol. 3 in the North Holland Series

Andrew, Alex M.
Feedback versus Autonomy
Review of value of autopoiesis for SOS (self-organizing systems).
Contrast with AI of the Feigenbaum school based on AI workers’ alleged view that there exist too many “a priori” structural features in the nervous system for general systemic approaches to explain.
Progress in Cybernetics and Systems Research, Vol. 9, Washington DC / London: Hemisphere, 1982, pp. 301-304.

Ayala, Francisco J.
Teleological Explanations in Evolutionary Biology
Philosophy of Science, March 1970, pp. 1 - 15.
A critical discussion of the reappearance of Aristotelian concepts of purpose in the context of evolution and biological systems. Ayala differentiates between causal and teleological explanations (cf. Evered, 1976), and insinuates that the distinction is one of observational perspective and contextual consistency. He claims that for 3 categories of biological phenomena teleological explanations are appropriate: (1) where the end-state is consciously anticipated by the agent (i.e., purposeful activity); (2) self-regulation with regard to one or more key variables; (3) “(I)n reference to

structures anatomically and physiologically designed to perform a certain function.” (p. 9)

Baert, Patrick, and Jan De Schampheleire
Autopoiesis, Self-organisation and Symbolic Interactionism: Some Convergences
Kybernetes, Vol. 17, no. 1 (1988), pp. 60 - 69.
A comparative analysis of autopoietic social theory and the “symbolic interactionism” of Mead / Blumer (sociology). Points of comparision include: environment / external determinants of behavior; social/self-consciousness; objectivity / subjectivity in perspectives (epistemology); openness/ democracy; novelty / emergence in human),

Benarz, John (Jr.)
Autopoiesis: The Organizational Closure of Social Systems
Systems Research, Vol. 5 (1988), no. 1, pp. 57 - 64.
An attempt to support Luhmann’s view of social systems as autopoietic unities.

Ben-Eli, Michael U.
Self-Organization, Autopoiesis, and Evolution
in Zeleny, M. (ed.), Autopoiesis: A Theory of Living Organization,
New York: North Holland, 1981.(Vol. 3 in the North Holland Series in General Systems Research)

Benseler, Frank, Peter M. Hejl, and Wolfram K. Kock (eds.)
Autopoiesis, Communication, and Society: The Theory of Autopoietic Systems in the Social Sciences
This is a collection of papers given at a conference in April 1979 on the theme of “The Theory of Autopoietic Systems as a New Foundation for the Social Sciences”.
Frankfurt: Campus Verlag, 1980

Blankenburg, Erhard
The Poverty of Evolutionism: A Critique of Teubner’s Case for “Reflexive Law”
cf. Teubner’s rejoinder in the same issue.
Law & Society Review, Vol. 18 (1984), no. 2, pp. 273-289.

Braten, Stein
The Third Position -- Beyond Artificial and Autopoietic Reduction
Kybernetes, Vol. 13 (1984), pp. 157 - 163.
This is one of the few attempts to apply autopoietic theory to a discussion of social systems. Unfortunately, Braten severely misuses the theory, applying it only with strict regard to “organizational closure” of social systems. His usage is therefore biased, and self-serving. He compares autopoietic systems to Leibniz’s monads, even though the application of this analogy is itself an indication of the observer’s role, which he completely ignores in his discussion.

Braten, Stein
Artifacts of Mind
in Christiansen, Bjorn (ed.), Psykologprofesjonen moto r 2000,
Bergen: Universitetsforlaget AS, 1987, pp. 543 - 557.
This is a broad, general discussion of IT issues. Braten identifies the 2 distinct camps of AI/cognitivism and phenomenological. He sees them as immiscible, and discusses the prospects for accounting for both in IT design.

Campbell, D., and R. Draper (eds.)
Applications of Systemic Family Therapy: The Milan Method
New York: Grune and Stratton, 1985.

Castoriadis, Cornelius
The Imaginary: Creation in the Social-Historical Domain
in Livingston, Paisley (ed.), Disorder and Order: Proceedings of the Stanford International Symposium (Sept. 14-16, 1981),
Saratoga CA: ANMA Libri, 1984, pp. 146 - 161.
This discussion proceeds from the foundation of society being proemial to the individual (seen as a social actor). This makes Castoriadis an exponent of the Luhmann/sys-referential view, wherein the system is primary and the individual must fit. He makes reference to Varela’s autonomy as an exemplar of how such social systems are realized.

de Solla Price, Derek J.

Automata and the Origins of Mechanism and Mechanistic Philosophy
Technology and Culture, Vol. 5 (1966), pp. 9 - 23.

Deggau, Hans-Georg
The Communicative Autonomy of the Legal System
in Teubner, Gunther, Autopoietic Law: A New Approach to Law and
Society, Berlin: Walter de Gruyter, 1988, pp. 128 - 151.

Dell, Paul F.
Beyond Homeostasis: Toward a Concept of Coherence
Family Process, Vol. 21 (1982), pp. 21 - 41.
A discussion of the limitations of homeostasis as a systems-theoretic concept, including a retrospective look at Bateson, Maturana, and the impacts they have had on Dell. He says in conclusion that his “coherence” is very similar to Maturana’s concept of structure determination for systems.

Dell, Paul F.
In Search of Truth: On the Way to Clinical Epistemology
Family Process, Vol. 21 (1982), pp. 407 - 414.
*incorrectly cited in Simon (1985). A review of Dell’s path into (among other things) autopoietic theory as an inspiration to his view of therapy.

Dell, Paul F., and Harold A. Goolishian
“Order through Fluctuation”: An Evolutionary Epistemology for Human Systems
Aust. Journal of Family Therapy, Vol. 2 (1981), no. 4, pp. 175 - 184.

Dias, R. W. M.
Autopoiesis and the Judicial Process
Rechtstheorie, Vol. 11 (1980), pp. 257 - 282.[Berlin: Duncker & Humblot]

Dupuy, Jean-Pierre
Shaking the Invisible Hand
in Livingston, Paisley (ed.), Disorder and Order: Proceedings of the Stanford International Symposium (Sept. 14-16, 1981),
Saratoga CA: ANMA Libri, 1984, pp. 129 - 145.
This is a discussion of the effect which Adam Smith’s “invisible hand” had on the history of ideas, in terms of economics and in politics as well. Dupuy’s presentation goes on to discuss Rousseau’s attempts to put the law above the individuals, so as to disengage it from their ability to manipulate. Dupuy thereby draws remarkably close to the view of Luhmann -- that a social system (seen in a moral / political perspective) should be an autonomous entity. Well-written and provocative.

Dupuy, Jean Pierre
On the Supposed Closure of Normative Systems
in Teubner, Gunther, Autopoietic Law: A New Approach to Law and Society,
Berlin: Walter de Gruyter, 1988, pp. 51 - 69.

Efran, Jay, and Michael D. Lukens
The World According to Humberto Maturana
The Family Therapy Networker, May-June 1985, pp. 23 - 28 / 72 - 75.

Faucheax, Claude, and Spyros Makridakis
Automation or Autonomy in Organizational Design
International Journal of General Systems, Vol. 5 (1979), pp. 213 - 220.

Gaines, Brian R.
Autopoiesis: Some Questions
in Zeleny, M. (ed.), Autopoiesis: A Theory of Living Organization,
New York: North Holland, 1981.(Vol. 3 in the North Holland Series in General Systems Research)
A healthy but skeptical list of questions concerning the value and strength of autopoietic theory.

Goolishian, Harold A., and L. Winderman
Constructivism, Autopoiesis and Problem Determined Systems
The Irish Journal of Psychology, Vol. 9 (1988), pp. 130-143.

Harnden, Roger J.

The Languaging of Models: The Understanding and Communication of Models with Particular Reference to Stafford Beer’s Cybernetic Model of Organization Structure
Systems Practice, Vol. 3 (1990), no. 3, pp. 289-302.

Hejl, Peter M., Wolfram K. Kock, and Gerhard Roth (eds.)
Wahrnehmung und Kommunikation
Frankfurt: Lang, 1978.
This is a collection of papers given at a conference in April 1979 on the theme of “The Theory of Autopoietic Systems as a New Foundation for the Social Sciences”.

Hejl, Peter M.
The Problem of a Scientific Description of Society
in Benseler, Frank, Peter M. Hejl, and Wolfram K. Kock (eds.),
Autopoiesis, Communication, and Society: The Theory of Autopoietic Systems in the Social Sciences Frankfurt: Campus Verlag, 1980, pp. 147 - 162.

Hejl, Peter M.
The Definition of System and the Problem of the Observer: The Example of the Theory of Society
in Roth, Gerhard, and Helmut Schwegler (eds.), Self-organizing Systems: An Interdisciplinary Approach
Frankfurt/New York: Campus Verlag, 1981, pp. 170 - 185.

Hejl, Peter M.
Towards a Theory of Social Systems: Self-Organization and Self- Maintenance, Self-Reference and Syn-Reference
in Ulrich, Hans, and Gilbert J. B. Probst (eds.), Self- Organization and Management of Social Systems: Insights, Promises, Doubts, and Questions,
Berlin: Springer-Verlag, 1984, pp. 60 - 78.
This is the only person I’ve found yet (July 1990) who sees autopoiesis’ application to social systems as a “bottom-up” process --- i.e., the organisms come first, and the social system derives from them. Very well written, concise, with explicitly laid out definitions.

Hoffman, L.
A Constructivist Postion for Family Therapy
The Irish Journal of Psychology, Vol. 9 (1988), pp. 110-129.

Hoffman, L.
Constructing Realities: An Art of Lenses
Family Proc., Vol. 29 (1990), 1. (Page 1 only? -- or issue no. 1?)

Hogfors, Marie
Idagsidan (column), Svenska Dagbladet, Friday 27 June 1986.
“I biologin finns bara nu” (Interview with Humberto Maturana)
Maturana was in Stockholm for a conference sponsored by “Institutet fr systemteoretisk beteendevetenskap” (Institute for System-Theoretical Behavioral Sciences” (??). He discusses “co- drift” (mutual influence between people) and uses the term “multi- versum” (multi-verse, in contrast to universe) to describe the world as a set of observer-dependent interpretations.

Jantsch, Erich
Autopoiesis: A Central Aspect of Dissipative Self-Organization
in Zeleny, M. (ed.), Autopoiesis: A Theory of Living
Organization, New York: North Holland, 1981.(Vol. 3 in the North Holland Series in General Systems Research)
Jantsch attempts to categorize autopoiesis as one subcomponent of a more general class of systems, essentially that of dissipative structures.

Jumarie, Guy
Towards a Mathematical Theory of Autopoiesis
Cybernetica, Vol. XXX, no. 8 (1987), pp. 59 - 89.
This is a rip-off. Jumarie in fact never refers to autopoiesis except in the title, and he makes no references to other authors in autopoietic theory. His model is a rigorous mathematical attempt to apply Shannon’s information theory to explain self- organizing systems. He proceeds from 2 very non-autopoietic notions: information as an invariant quantum a la Shannon, and an “objective” for each system (shades of Aristotle’s teleological excesses!!).

Katakis, Charis
The Self-Referential Conceptual System: Towards an Operational Definition of Subjectivity
Systems Research, Vol. 7 (1990), no. 2, pp. 91-102.

Kauffman, Louis H., and Francisco J. Varela
Form Dynamics
Journal of Social Biological Structures, Vol. 3 (1980), pp. 171 - 206.

Keeney, Bradford P.
Ecosystemic Epistemology: An Alternative Paradigm for Diagnosis
Family Process, Vol. 18, no. 2 (June 1979), pp. 117 - 129.

Kenneally, Peter
Talking about Autopoiesis -- Order from Noise?
in Teubner, Gunther, Autopoietic Law: A New Approach to Law and Society,
Berlin: Walter de Gruyter, 1988, pp. 349 - 368.

Kenny, V., and G. Gardner
Constructions of Self-Organising Systems
The Irish Journal of Psychology, Vol. 9 (1988), pp. 1-24.

Kock, Wolfram K.
Autopoiesis and Communication
in Benseler, Frank, Peter M. Hejl, and Wolfram K. Kock (eds.),
Autopoiesis, Communication, and Society: The Theory of Autopoietic Systems in the Social Sciences
Frankfurt: Campus Verlag, 1980, pp. 87 - 112.

Kock, Wolfram K.
On Communication and the Stability of Social Systems
in Roth, Gerhard, and Helmut Schwegler (eds.), Self-organizing Systems: An Interdisciplinary Approach
Frankfurt/New York: Campus Verlag, 1981, pp. 145 - 169.

Ladeur, Karl-Heinz
Perspectives on a Post-Modern Theory of Law: A Critique of Niklas Luhmann, ‘The Unity of the Legal System’
in Teubner, Gunther, Autopoietic Law: A New Approach to Law and
Society, Berlin: Walter de Gruyter, 1988, pp. 242 - 282.

Locker, Alfred
Metatheoretical Presuppositions for Autopoiesis
in Zeleny, M. (ed.), Autopoiesis: A Theory of Living
Organization, New York: North Holland, 1981.(Vol. 3 in the North
Holland Series in General Systems Research)
A generally critical view of autopoietic theory. He closes with the complaint that Maturana is guilty of “scientism”, and that such approaches cannot be fruitful.

Lfgren, Lars
Life as an Autolinguistic Phenomenon
in Zeleny, M. (ed.), Autopoiesis: A Theory of Living
Organization, New York: North Holland, 1981.(Vol. 3 in the North
Holland Series in General Systems Research)

Luhmann, Niklas
The World Society as a Social System
International Journal of General Systems, Vol. 8 (1982), pp. 131 - 138.
A general commentary on social systems. Not too detailed with respect to his use of autopoiesis; not as detailed as his work on the legal system. He claims here that the social system is closed and autopoietic with respect to communicative acts (so maybe this is where he started the line of thought promoted by Bednarz).

Luhmann, Niklas
The Unity of the Legal System
in Teubner, Gunther, Autopoietic Law: A New Approach to Law and
Society, Berlin: Walter de Gruyter, 1988, pp. 12 - 35.

Luhmann, Niklas
Closure and Openness: On Reality in the World of Law
in Teubner, Gunther, Autopoietic Law: A New Approach to Law and
Society, Berlin: Walter de Gruyter, 1988, pp. 335 - 348.

Maturana, Humberto R.
The Neurophysiology of Cognition
in Garvin, P. (ed.), Cognition: A Multiple View, New York:
Spartan Books, 1969, pp. 3 - 24.
This is cited by Varela (1979) as being a particularly important early paper. It is a clearly written, concise introduction to the direction Maturana was going in 1969 -- long before he and his colleagues devised the terminology of “organization”, “structure”, and “autopoiesis”. It contains his only discussion of AI -- computer systems as autonomous systems of the same type as his analysis of living ones. A remarkably satisfying paper; terminological disjunction aside, it is one of the best papers a beginner can read.

Maturana, Humberto R.
The Organization of the Living: A Theory of the Living
Organization
International Journal of Man-Machine Studies, Vol. 7 (1975), pp.
313 - 332.

Maturana, Humberto R.
Communication and Representation Functions
in Piaget, J. (ed.), Encyclopedie de la Pleaide, Paris:
Gallimard, 1975.

Maturana, Humberto R.
Cognition
in Hejl, Peter M., Wolfram K. Kock, and Gerhard Roth
(eds.),Wahrnehmung und Kommunikation, Frankfurt: Lang, 1978, pp. 29 - 49.

Maturana, Humberto R.
Biology of Language: The Epistemology of Reality
in Miller, George A., and Elizabeth Lenneberg (eds.), Psychology and Biology of Language and Thought: Essays in Honor of Eric
Lenneberg, New York: Academic Press, 1978, pp. 27-63.
Not Maturana’s best written essay, but one in which he concentrates on his view of linguistic interactions. There is some terminology used/defined here that does not get such treatment elsewhere. Highly recommended to finish off (but certainly not to begin) one’s exposure to Maturana.

Maturana, Humberto R.
Man and Society
in Benseler, Frank, Peter M. Hejl, and Wolfram K. Kock (eds.),
Autopoiesis, Communication, and Society: The Theory of Autopoietic Systems in the Social Sciences
Frankfurt: Campus
Verlag, 1980, pp. 11 - 32.

Maturana, Humberto R.
Autopoiesis
in Zeleny, M. (ed.), Autopoiesis: A Theory of Living
Organization, New York: North Holland, 1981.(Vol. 3 in the North Holland Series in General Systems Research)

Maturana, Humberto R.
What is It to See?
Arch. Biol. Med. Exp. (full title unknown), Vol. 16 (1983), pp. 255-269.

Maturana, Humberto R.
The Mind is Not in the Head
Journal of Social and Biological Structures, Vol. 8 (1985), pp. 303-311.

Maturana, Humberto R.
Ontology of observing: The biological foundations of self consciousness and the

physical domain of existence
in the conference workbook for “Texts in Cybernetic Theory”, an In
Depth Exploration of the Thought of Humberto R. Maturana, William T. Powers, and Ernst von Glasersfeld, American Society of Cybernetics, Felton CA, October 18-23, 1988.

Maturana, Humberto R.
Reality: The Search for Objectivity or the Quest for a Compelling Argument
The Irish Journal of Psychology, Vol. 9 (1988), no. 1, pp. 25-82.

Maturana, Humberto R., and Gloria D. Guiloff
The Quest for the Intelligence of Intelligence
Journal of Social and Biological Structures, Vol. 3 (1980), pp. 135 - 148.

Maturana, Humberto R., G. Uribe, and Samy G. Frenk
A Biological Theory of Relativistic Colour Coding in the Primate Retina
A discussion of nervous system closure with reference to certain visual effects.
Arch. Biol. Med. Exp. Supl. (full title unknown), Vol. 1 (1968), pp. 1-30.

Maturana, Humberto R., Francisco Varela, and Samy G. Frenk
Size Constancy and the Problem of Perceptual Spaces
Cognition, Vol. 1 (1972), no. 1, pp. 97 - 104.
A discussion of nervous system closure with reference to certain visual effects.

Maturana, Humberto R., and Francisco J. Varela
Autopoiesis and Cognition: The Realization of the Living
Boston Studies in the Philosophy of Science [Cohen, Robert S., and Marx W. Wartofsky (eds.)], Vol. 42,
Dordecht (Holland): D. Reidel Publishing Co., 1980.

Maturana, Humberto R., and Francisco J. Varela
The Tree of Knowledge: The Biological Roots of Human Understanding
Boston: New Science Library, 1987.
This is the most accessible introduction to their theories.
NOTE: New edition to be issued in 1992 (source: Library of
Congress database)

Mendez, C., F. Coddou, and Humberto R. Maturana
The Bringing Forth of Pathology
The Irish Journal of Psychology, Vol. 9 (1988),pp. 144-172.

Mingers, John
An Introduction to Autopoiesis -- Implications and Applications
Systems Practice, Vol. 2 (1989), no. 2, pp. 159 - 180.

Mingers, John
An Introduction to Autopoiesis: A Reply to Fenton Robb’s Comment
Systems Practice, Vol. 2 , no.3 (September 1989), pp. 349-352.
F. Robb (after welcoming Mingers’ article) suggests that social systems may indeed be autopoietic, and he invites Mingers to be less skeptical about this issue. Mingers lists three conditions for autopoiesis (self-production, boundary formation, and lack of a need for adaptation for social systems) to be applied to social systems, none of which he believes hold.

Mingers, John
The Philosophical Implications of Maturana’s Cognitive Theories
Systems Practice, Vol. 3 (1990), no. 6, pp. 569-584.

Mingers, John
The Cognitive Theories of Maturana and Varela
Systems Practice, Vol. 4 (1991), no. 4, pp. 319-338.
A straightforward look at the phenomenological aspects of M & V’s theories. This is somehow neither as penetrating nor as comprehensive as Mingers’ analysis of the philosophical aspects (1990). However, it is certainly a good concise intro to the cognitive aspects of the theories and an overview of its application into 3 fields (family therapy, computer design, and law), which are covered in more detail elsewhere.

Nissan, Ephraim

Knowledge Acquisition and Metarepresentation: Attribute Autopoiesis Methodologies for Intelligent Systems. Proceedings of the Second International Symposium,
New York: Elsevier (North Holland), 1987, pp. 240 - 247.

Ost, Francois
Between Order and Disorder: The Game of Law
in Teubner, Gunther, Autopoietic Law: A New Approach to Law and Society,
Berlin: Walter de Gruyter, 1988, pp. 70 - 96.

Pask, Gordon
Organizational Closure of Potentially Conscious Systems
in Zeleny, M. (ed.), Autopoiesis: A Theory of Living Organization,
New York: North Holland, 1981.(Vol. 3 in the North Holland Series in General Systems Research)
Pask's explanation of the relations between autopoiesis and his work in conversational systems. This is frankly the most accessible outline of his work I have read.

Pattee, Howard H.
The Nature of Hierarchical Controls in Living Matter
in Rosen, Robert (ed.), Foundations of Mathematical Biology,
New York: Academic Press, 1972, pp. 1 - 22.
Center for Theoretical Biology, SUNY at Buffalo, Amherst NY

Ravn, Inger, and Thomas Soderqvist
Having It Both Ways: Some Critical Comments on Humberto Maturana's Contribution to the Constructivist Movement in Cybernetics
Paper submitted to Cybernetics, 1988. No known publication date. A fuzzy critique of HM, primarily centered on the authors' discontent with HM's apparent "closure" (arrogance?) in some personal appearances. On a more concrete level, they bemoan Maturana's dilemma of how to apply scientific/ rational/ biological data to support an essentially subjective theory. This is worth some further enquiry, but these authors do not provide it. All in all, a surprisingly pathetic paper. They do mention the applications of autopoietic theory in family therapy in Scandinavia: apparently Maturana's prime focus in (1) Scandinavia and (2) the 1980's.

Robb, Fenton F.
Cybernetics in Management Thinking
Systems Research, Vol. 1, no. 1 (1984), pp. 5 - 23.

Robb, Fenton F.
Towards a 'Better' Scientific Theory of Human Organization
Journal of the Operational Research Society, Vo. 36 (1985), no. 6, pp. 463 - 466.

Robb, Fenton F.
The limits to human organization: the emergence of autopoietic systems
in Jackson, M. C., P. Keys, and S. A. Cropper (eds.), Operational
Research and the Social Sciences, New York / London: Plenum, 1989, pp. 247-251.

Robb, Fenton F.
Cybernetics and Suprahuman Autopoietic Systems
Systems Practice, Vol. 2, no. 1 (March 1989), pp. 47 - 74.
Robb is of the opinion that (1) social systems can indeed be considered autopoietic, thus putting him in agreement with (e.g.) Luhmann, Bednarz. He suggests that there may exist "suprahuman" autopoietic systems which are emerging from large corporate, government, and other systems undergoing stressful conditions which have driven them far from equilibrium (his reasoning is by analogy with respect to Prirogrine's "dissipative structures"). He further suggests that such suprahuman systems would exhibit as their primary controlling motive (goal) their own maintenance, regardless (and perhaps injurious) of humans associated with them. This can be seen as a critically crafted conspiracy theory.

Robb, Fenton F.
The Application of Autopoiesis to Social Organizations -- A Comment on John Mingers' "An Introduction to Autopoiesis: Implications and Applications"
Systems Practice, Vol. 2, no. 3 (September 1989), pp. 343-348.
Cf. Robb's position on suprahuman autopoietic systems. He claims that Mingers' reservations on the applicability of autopoiesis to social systems can be overcome, although he acknowledges the difficulty of the issues involved. His rebuttal is done

without specific support from examples or arguments, and maintains the tone of an admonition more than that of a compelling rejoinder.

Robb, Fenton F.
The Application of Autopoiesis to Social Organizations -- A Comment on John Mingers' Reply
Systems Practice, Vol. 2, no. 3 (September 1989), pp. 353-360.

Robb, Fenton F.
Accounting -- A Virtual Autopoietic System?
Systems Practice, Vol. 4 (1991), no. 3, pp. 215 - 235.

Roth, Gerhard, and Helmut Schwegler (eds.)
Self-organizing Systems: An Interdisciplinary Approach
Frankfurt/New York: Campus Verlag, 1981.

Roth, Gerhard
Conditions of Evolution and Adaptation in Organisms as AutopoieticSystems
in Mossakowski, D., and G. Roth (eds.), Environmental Adaptation and Evolution,
Stuttgart / New York: Gustav Fischer, 1982, pp. 37 - 48.

Sawada, Daiyo, and Michael T. Caley
Recursive Complementarity in the Cybernetics of Education Cybernetica, Vol. XXIX. no. 2 (1986), pp. 93 - 104.
The authors discuss 3 versions of cybernetics (C-I, II, and III). They claim CIII can be achieved by recursive, complementary switching between CI and CII modalities. They claim that this switching is a natural occurrence in autopoietic living systems as defined by M & V. CI = control via negative feedback; CII = extension via positive feedback. They claim evolution is a recursive interplay between these two.

Simon, Richard
A Frog's Eye View of the World
The Family Therapy Networker, May/June 1985. A review of Maturana's theories as applied to family therapy. This includes an interview with Maturana himself, in which there are some key points bluntly stated. Simon cites systems theorist Paul Dell as the one "...largely responsible for drawing family therapists' attention to Maturana's work..."

Teubner, Gunther
Autopoiesis in Law and Society: A Rejoinder to Blankenburg
Law & Society Review, Vol. 18 (1984), no. 2, pp. 291 - 301.
A response to Erhard Blankenburg's critique of Teubner's autopoietic, evolutionary views on law (following Luhmann). Blankenburg's article appears in the same issue.

Teubner, Gunther (ed.)
Autopoietic Law: A New Approach to Law and Society
Berlin / New York: Walter de Gruyter, 1988.

Teubner, Gunther
Evolution of Autopoietic Law
in Teubner, Gunther, Autopoietic Law: A New Approach to Law and Society,
Berlin: Walter de Gruyter, 1988, pp. 217 - 243.

Ulanowicz, Robert E.
A Phenomenology of Evolving Networks
Systems Research, Vol. 6 (1989), no. 3, pp. 209-217.

Ulrich, Hans, and Gilbert J. B. Probst (eds.)
Self-Organization and Management of Social Systems: Insights, Promises, Doubts, and Questions
Berlin: Springer-Verlag, 1984.

Uribe, Ricardo B.
Modeling Autopoiesis
in Zeleny, M. (ed.), Autopoiesis: A Theory of Living Organization,
New York: North Holland, 1981.(Vol. 3 in the North Holland Series in General Systems Research)

Varela, Francisco
A Calculus for Self-Reference
International Journal of General Systems, Vol. 2 (1975), pp. 5-24.

Varela, Francisco J.
Not One, Not Two
CoEvolution Quarterly, Fall 1976, pp. 62 - 67.
It is basically a discussion of material on complementary duals which can be found reflected in Varela (1979) Principles of Biological Autonomy.

Varela, Francisco
On Being Autonomous: The Lessons of Natural History for Systems Theory in Klir, George J. (ed.), Applied General Systems Research: Recent Developments and Trends, New York: Plenum, 1978, pp. 77 - 84.

Varela, Francisco J.
Principles of Biological Autonomy
New York: Elsevier (North Holland), 1979.

Varela, Francisco J.
Describing the Logic of the Living
in Zeleny, M. (ed.), Autopoiesis: A Theory of Living Organization,
New York: North Holland, 1981.(Vol. 3 in the North Holland Series in General Systems Research)

Varela, Francisco J.
Autonomy and Autopoiesis
in Roth, Gerhard, and Helmut Schwegler (eds.), Self-organizing Systems: An Interdisciplinary Approach
Frankfurt/New York: Campus Verlag, 1981, pp. 14 - 23.

Varela, Francisco J.
Two Principles for Self-Organization
in Ulrich, Hans, and Gilbert J. B. Probst (eds.), Self-
Organization and Management of Social Systems: Insights, Promises, Doubts, and Questions,
Berlin: Springer-Verlag, 1984, pp. 25 - 32.

Varela, Francisco J.
Living Ways of Sense-Making: A Middle Path for Neuroscience
in Livingston, Paisley (ed.),
Disorder and Order: Proceedings of the Stanford International Symposium (Sept. 14-16, 1981),
Saratoga CA: ANMA Libri, 1984, pp. 208 - 224.

Varela, Francisco J.
Structural coupling and the origin of meaning in a simle cellular automata
in: Secarz, E., et al. (eds.) The Semiotics of Cellular Communications in the Immune System
New York: Springer-Verlag, 1988, pp. ??????

Varela, Francisco J.
Reflections on the circulation of concepts between a biology of cognition and systemic family family therapy
Family Process, Vol. 28 (1989), no. 1 (March), pp. 15 - 24.
This is Varela’s discrete way of drawing back from the application of autopoietic theory to social systems in general (and family systems in particular) -- a point on which both he and Maturana have been far more reluctant to proceed than some others (e.g., N. Luhmann).

Varela, Francisco J., and D. Johnson
On Observing Natural Systems
CoEvolution Quarterly, Summer 1976, pp. 26.

Varela, Francisco, and Joseph A. Goguen
The Arithmetic of Closure
Journal of Cybernetics, Vol. 8 (1978), pp. 291-324.

Varela, Francisco J., Humberto R. Maturana, and R. Uribe
Autopoiesis: The Organization of Living Systems, Its Characterization and a Model
Biosystems 5 (1974), pp. 187 - 196. [Amsterdam: North-Holland]

Varela, Francisco, and Humberto Maturana
Mechanism and Biological Explanation
Philosophy of Science, Vol. 39 (1972), pp. 378 - 382.
A critical look at the applicability of machine models to explain living systems. This paper was received in Jan. 1970, and represents a very early view of the concepts of organization and structure. In fact (p. 378) it appears that they originally termed "organization" as "structure or theory of the machine".

Varela, Francisco J., Evan Thompson, and Eleanor Rosch
The Embodied Mind: Cognitive Science and Human Experience
Cambridge MA: MIT Press, 1991

Vaz, N. M., and Francisco J. Varela
Self and Non-Sense: An Organism-Centered Approach to Immunology
(1979) "Principles of Biological Autonomy".
Medical Hypotheses, Vol. 4 (1978), pp. 231 - 267.
This is the actual paper which describes the autopoietic re- interpretation of the immune system which is summarized in Varela's

von Glasersfeld, Ernst
The Concept of Equilibration in a Constructivist Theory of Knowledge
in Benseler, Frank, Peter M. Hejl, and Wolfram K. Kock (eds.),
Autopoiesis, Communication, and Society: The Theory of Autopoietic Systems in the Social Sciences
Frankfurt: Campus Verlag, 1980, pp. 75 - 86.

von Glasersfeld, Ernst
An Epistemology for Cognitive Systems
in Roth, Gerhard, and Helmut Schwegler (eds.), Self-organizing Systems: An Interdisciplinary Approach
Frankfurt/New York: Campus Verlag, 1981, pp. 121 - 131.

West, David M., and Larry E. Travis
The Computational Metaphor and Artificial Intelligence: A Reflective Examination of a Theoretical Falsework
AI Magazine, Spring 1991, pp. 64-79.

West, David M., and Larry E. Travis
From Society to Landscape: Alternative Metaphors for Artificial Intelligence
AI Magazine, Summer 1991, pp. 69-83.
This is a horribly distorted and misinformed attempt to characterize autopoietic theory and apply it as an alternative AI paradigm. Useful as a punching bag, and nothing else.

Zeleny, Milan, and Norbert A. Pierre
Simulation of Self-Renewing Systems
in Jantsch, Eric, and Conrad H. Waddington (eds.), Evolution and Consciousness: Human Systems in Transition,
Reading MA: Addison-Wesley, 1976.
Provides a model for simulating autopoietic systems via computer.

Zeleny, Milan
Self-Organization of Living Systems: A Formal Model of Autopoiesis
International Journal of General Systems, Vol. 4 (1977), pp. 13 - 28.
Description of Zeleny's computerized simulation of chemical bonding as a demonstration of an autopoietically-inspired model. See also Zeleny & Pierre.

Zeleny, Milan
Special Book Reviews
International Journal of General Systems, Vol. 5 (1979), pp. 63- 71.
This is a more detailed explanation of the precursors Zeleny had previously cited for Maturana's autopoietic theory (in the paper "Self-Organization of Living Systems: A Formal Model of Autopoiesis", International Journal of General Systems, Vol. 3 (1977), pp. 13 - 28), including brief discussion of the points of correspondence between them

and autopoiesis.

Zeleny, Milan (ed.)
Autopoiesis, Disspative Structures, and Spontaneous Social Orders
AAAS Selected Symposium 55 (AAAS National Annual Meeting,
Houston TX, Jan. 3-8, 1979), Boulder CO: Westview Press, 1980.

Zeleny, Milan (ed.)
Autopoiesis: A Theory of Living Organization
New York: North Holland, 1981.(Vol. 3 in the North Holland Series in General Systems Research)

Zeleny, Milan
What is Autopoiesis?
in Zeleny, M. (ed.), Autopoiesis: A Theory of Living Organization,
New York: North Holland, 1981.(Vol. 3 in the North Holland Series in General Systems Research)

Zeleny, Milan
Spontaneous Social Orders
International Journal of General Systems, Vol. 11 (1985), pp. 117 - 131.
This is Zeleny’s discussion of self-organization in social relations/groups, etc. It includes a historical review of writers which Zeleny links to autopoietic thought in terms of their addressing social self-organization.

Zeleny, Milan
Management Support Systems: Towards Integrated Knowledge
Management
New York: Elsevier (North-Holland), Human Systems Management,
Vol. 7 (1987), pp. 59 - 70.
A critique of KBS applied to management support. Emphasis on the non-rationalistic nature of “knowledge” -- “not a processing of information, but a coordination of action”. Many parallels with “The Automated Expert”. Suggests that a re-integrative phase (opposite of division of labor) is occurring, and generally suggests that such to-and-fro developments are the way large social orders evolve. Can this be tied to the sys- / syn- dualism??

Zeleznikar, Anton P.
Information Determinations I
Cybernetica, Vol. XXXI, no. 3 (1988), pp. 182 - 213.
A review of the concept of “information”, primarily drawing on lexical references.

Apeiron Press

PO Box 4402
Garden Grove,
California 92842-4402

714-638-7376
714-638-1210
palmer@think.net
palmer@netcom.com
palmer@exo.com
Dataline 714-638-0876

Draft #1 950710 Editorial Copy.
Not for distribution.

This book was set using Framemaker document publishing software by the author.

Electronic Version in Adobe Acrobat PDF available at http://server.snni.com:80/~palmer/homepage.html

Library of Congress
Cataloging in Publication Data

Palmer, Kent Duane

WILD SOFTWARE META-SYSTEMS

Bibliography
Includes Index

1. Philosophy-- Ontology
2. Software Engineering
3. Software Design Methods

I. Title

[XXX000.X00 199x]
9x-xxxxx
ISBN 0-xxx-xxxxx-x

Keywords:

Software, Design Methods, Ontology,
Integral Software Engineering
Methodology, Systems Theory